

FIG. 1

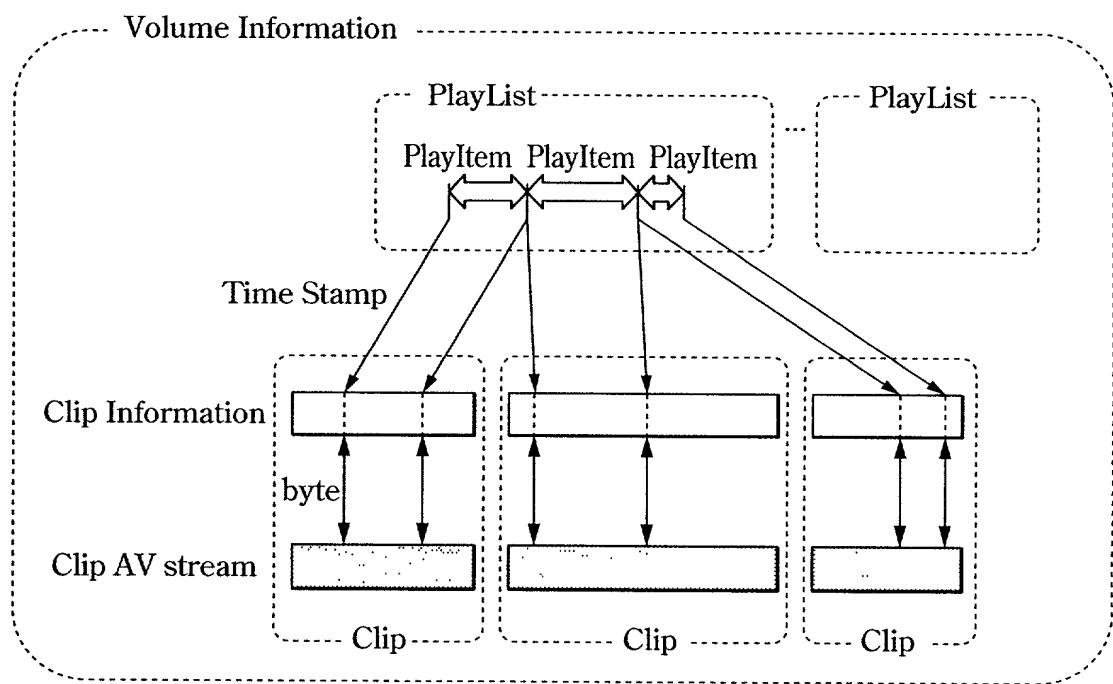
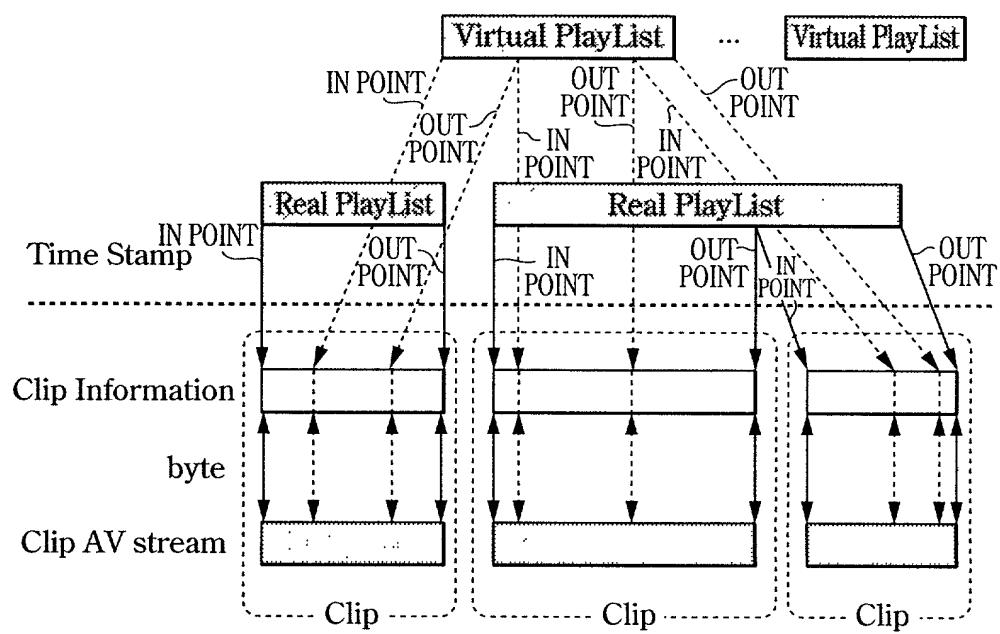


FIG.2

**FIG.3**

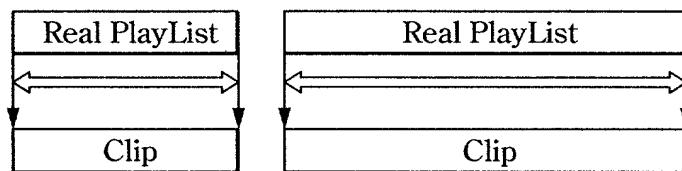


FIG.4A

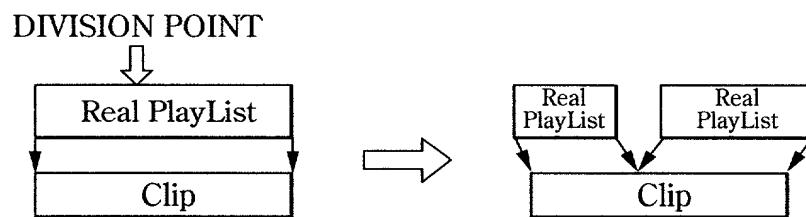


FIG.4B

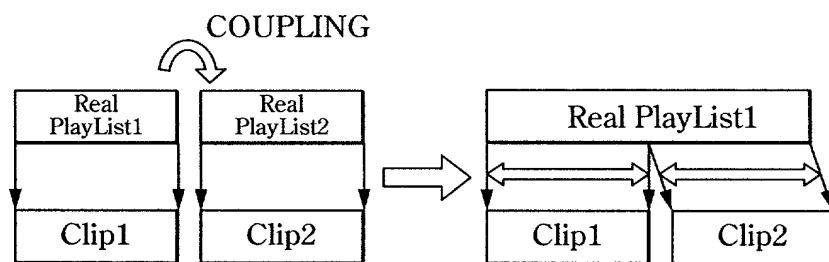


FIG.4C

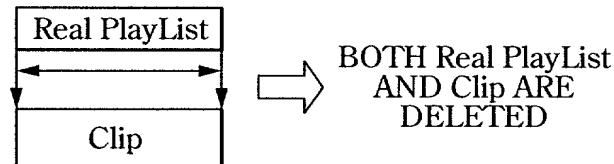


FIG.5A

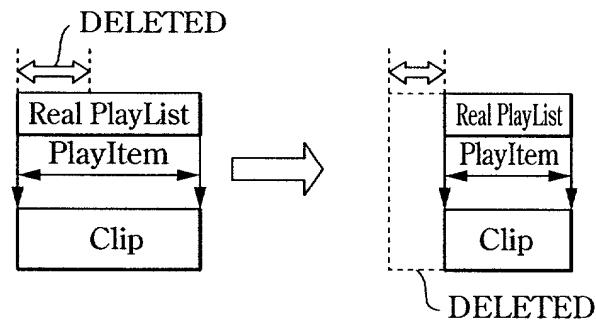


FIG.5B

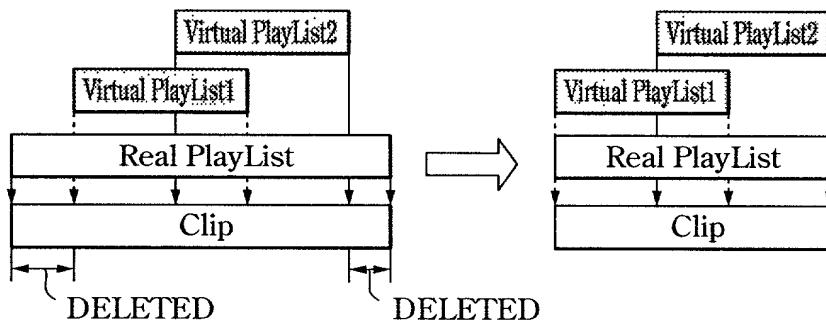
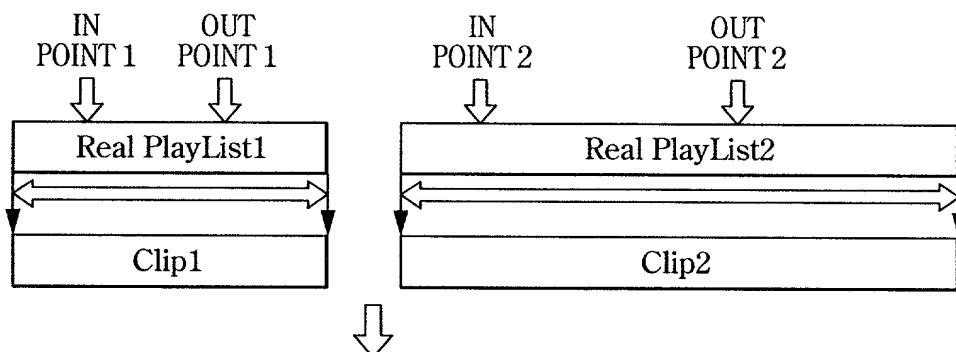
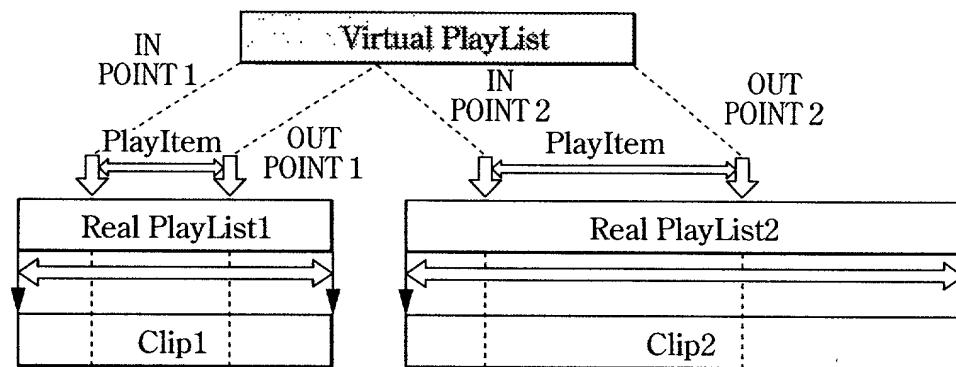
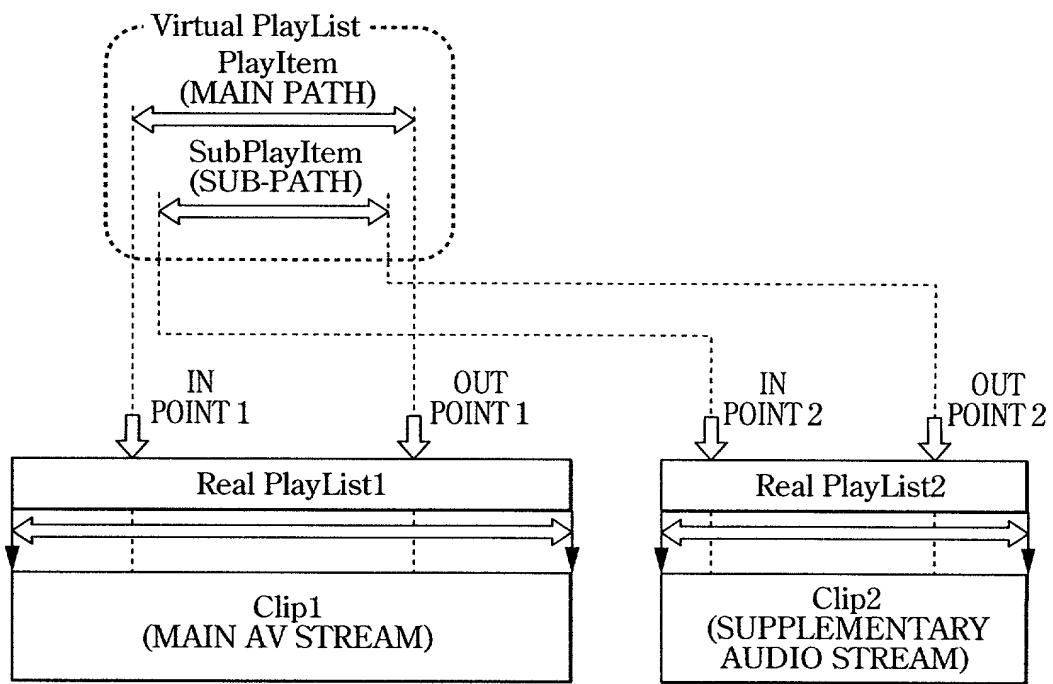


FIG.5C

FIG.6A**FIG.6B**

**FIG.7**

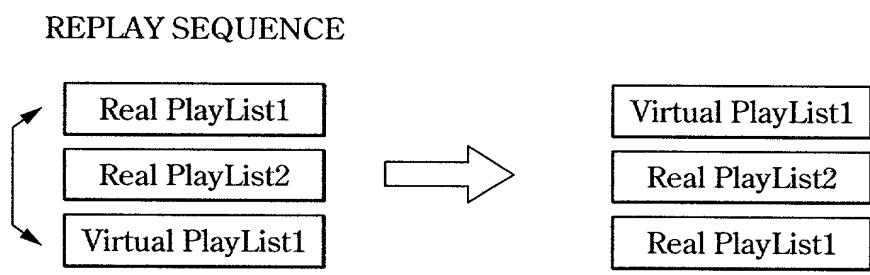


FIG.8

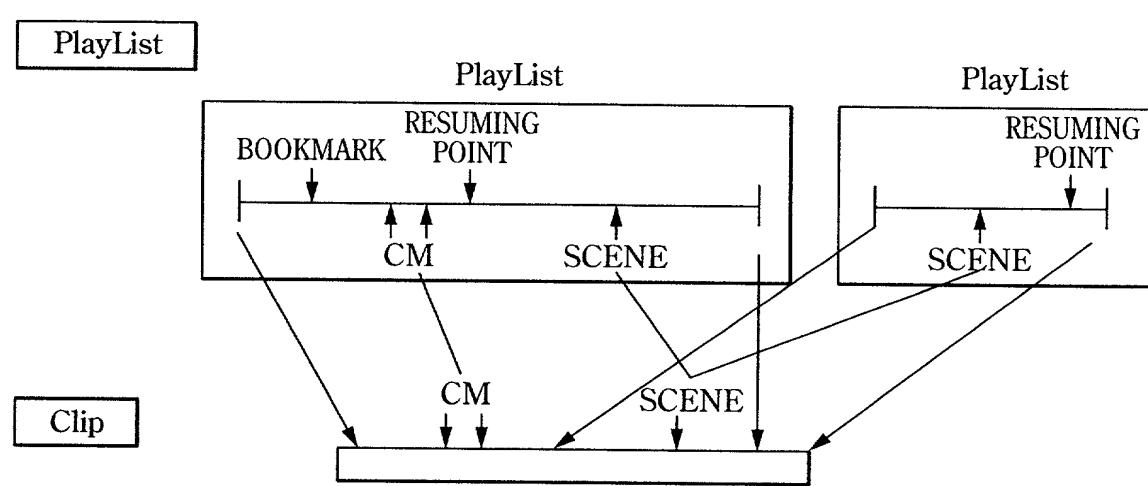


FIG.9

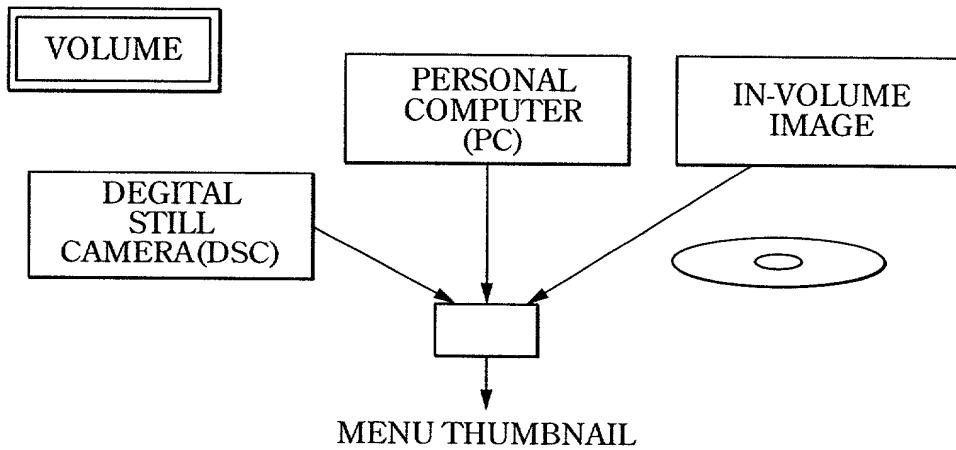


FIG.10

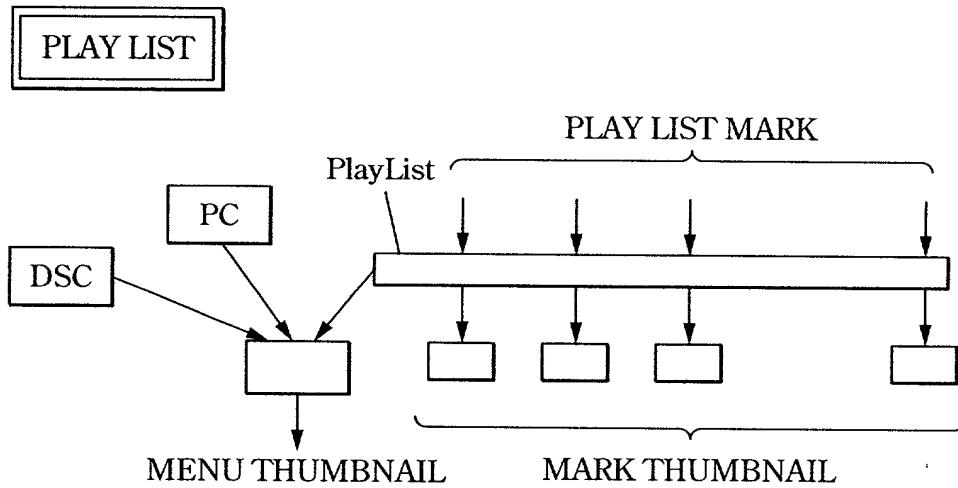


FIG.11

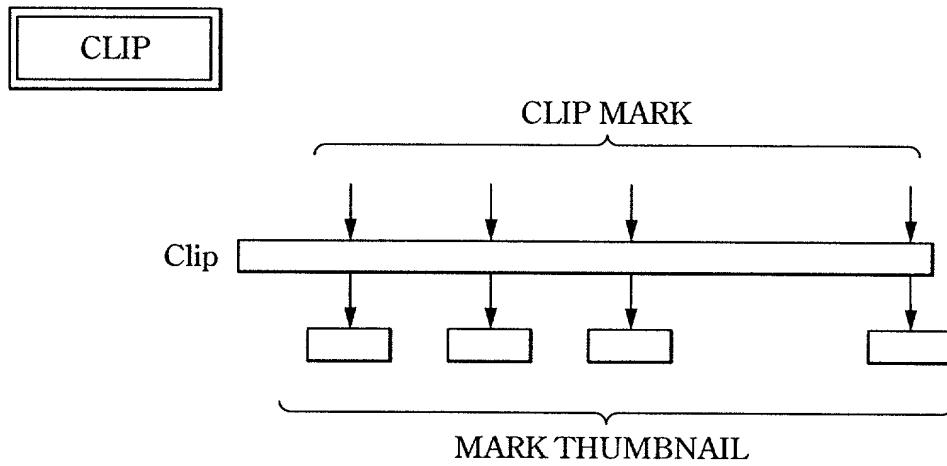


FIG.12

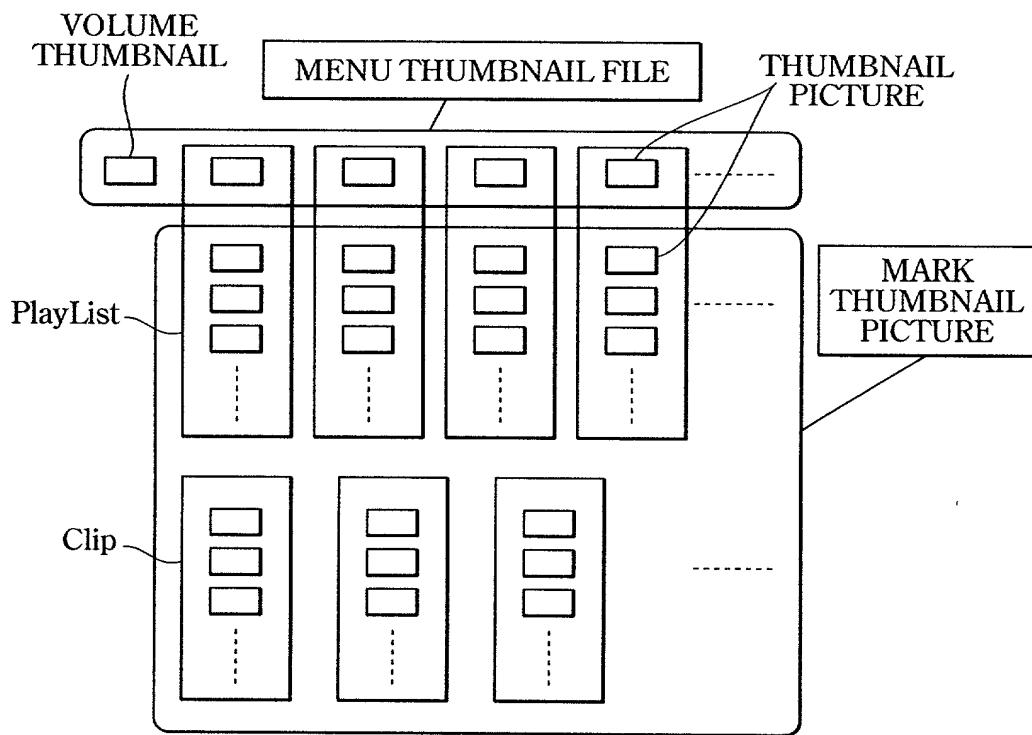


FIG.13

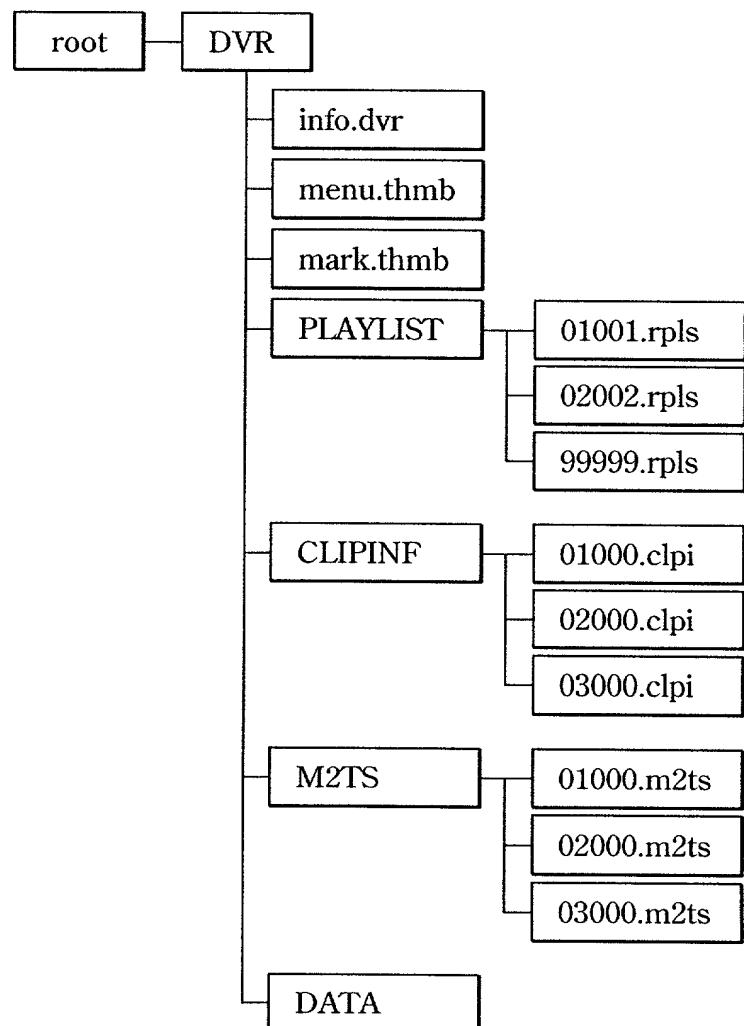


FIG.14

SYNTAX	NUMBER OF BYTES	ABBREVIATION
info.dvr {		
TableOfPlayLists_Start_address	32	uimsbf
MakersPrivateData_Start_address	32	uimsbf
reserved	192	bslbf
DVRVolume()		
for (i=0;i<N1;i++){		
padding_word	16	bslbf
}		
TableOfPlayLists()		
for (i=0;i<N2;i++){		
padding_word	16	bslbf
}		
MakersPrivateData()		
}		

FIG.15

SYNTAX	NUMBER OF BYTES	ABBREVIATION
DVRVolume{		
version_number	8*4	bslbf
length	32	uimsbf
ResumeVolume()		
UIAppInfoVolume()		
}		

FIG.16

SYNTAX	NUMBER OF BYTES	ABBREVIATION
ResumeVolume()		
reserved	15	bslbf
valid_flag	1	bslbf
resume_PlayList_name	8*10	bslbf
}		

FIG.17

SYNTAX	NUMBER OF BYTES	ABBREVIATION
UIAppInfoVolume{		
character_set	8	bslbf
name_length	8	uimsbf
Volume_name	8*256	bslbf
reserved	15	bslbf
Volume_protect_flag	1	bslbf
PIN	8*4	bslbf
ref_thumbnail_index	16	uimsbf
reserved_for_future_use	256	bslbf
}		

FIG.18

VALUE	CHARACTER LETTER ENCODING
0x00	Reserved
0x01	ISO/IEC 646 (ASCII)
0x02	ISO/IEC 10646-1 (Unicode)
0x03-0xff	Reserved

FIG.19

SYNTAX	NUMBER OF BYTES	ABBREVIATION
TableOfPlayLists(){		
version_number	8*4	bslbf
length	32	uimsbf
number_of_PlayLists	16	uimsbf
for (i=0; i< <i>number_of_PlayLists</i> ; i++){		
PlayList_file_name	8*10	bslbf
}		
}		

FIG.20

SYNTAX	NUMBER OF BYTES	ABBREVIATION
TableOfPlayLists()		
version_number	8*4	bslbf
length	32	uimsbf
number_of_PlayLists	16	uimsbf
for (i=0; i< number_of_PlayLists ; i++) {		
PlayList_file_name	8*10	bslbf
UIAppInfoPlayList()		
}		
}		

FIG.21

SYNTAX	NUMBER OF BYTES	ABBREVIATION
MakersPrivateData()		
version_number	8*4	bslbf
length	32	uimsbf
if (length !=0){		
mpd_blocks_start_address	32	uimsbf
number_of_maker_entries	16	uimsbf
mpd_block_size	16	uimsbf
number_of_mpd_blocks	16	uimsbf
reserved	16	bslbf
for (i=0; i<number_of_maker_entries; i++){		
maker_ID	16	uimsbf
maker_model_code	16	uimsbf
start_mpd_block_number	16	uimsbf
reserved	16	bslbf
mpd_length	32	uimsbf
}		
stuffing_bytes	8*2*L1	bslbf
for(j=0; j<number_of_mpd_blocks; j++){		
mpd_block	mpd_block_size*1024*8	
}		
}		
}		

FIG.22

SYNTAX	NUMBER OF BYTES	ABBREVIATION
xxxxx.rpls / yyyy.ypls {		
PlayListMark_Start_address	32	uimsbf
MakersPrivateData_Start_address	32	uimsbf
reserved	192	bslbf
PlayList()		
for (i=0;i<N1;i++){		
padding_word	16	bslbf
}		
PlayListMark()		
for (i=0;i<N2;i++){		
padding_word	16	bslbf
}		
MakersPrivateData()		
}		

FIG.23

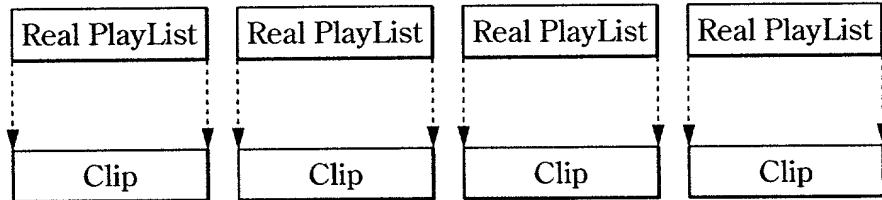


FIG.24A

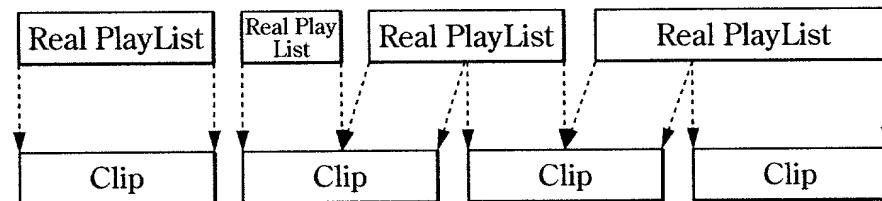


FIG.24B

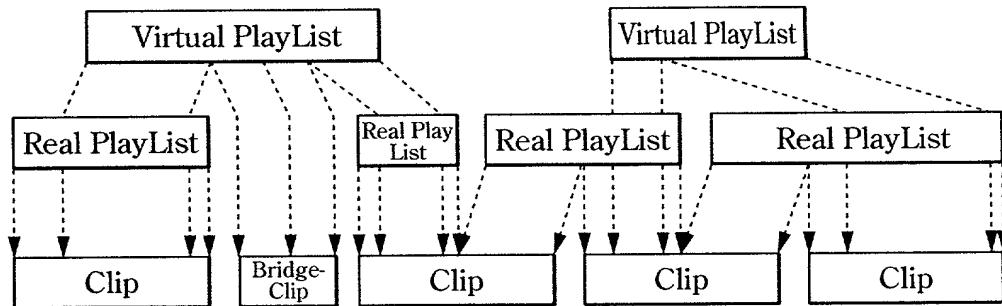


FIG.24C

SYNTAX		NUMBER OF BYTES	ABBREVIATION
PlayList()			
version_number	8*4	bslbf	
length	32	uimsbf	
PlayList_type	8	uimsbf	
CPI_type	1	bslbf	
reserved	7	bslbf	
UIAppInfoPlayList()			
number_of_PlayItems // main path	16	uimsbf	
if (<Virtual PlayList>){			
number_of_SubPlayItems // sub path	16	uimsbf	
else{			
reserved	16	bslbf	
}			
for (PlayItem_id=0;			
PlayItem_id<number_of_PlayItems;			
PlayItem_id++) {			
PlayItem() //main path			
}			
if (<Virtual PlayList>){			
if (CPI_type==0 && PlayList_type==0){			
for (i=0; i<number_of_SubPlayItems; i++)			
SubPlayItem() //sub path			
}			
}			
}			

FIG.25

PlayList_type	MEANING
0	PLAY LIST FOR AV RECORDING ALL CLIPS REFERENCED IN THIS PLAY LIST MUST CONTAIN ONE OR MORE VIDEO STREAMS
1	PLAY LIST FOR AUDIO RECORDING ALL CLIPS REFERENCED IN THIS PLAYLIST MUST CONTAIN ONE OR MORE AUDIO STREAMS AND MUST NOT CONTAIN VIDEO STREAMS
2-255	reserved

FIG.26

SYNTAX	NUMBER OF BYTES	ABBREVIATION
UIAppInfoPlayList20{		
character_set	8	bslbf
name_length	8	uimsbf
PlayList_name	8*256	bslbf
reserved	8	bslbf
record_time_and_date	4*14	bslbf
reserved	8	bslbf
duration	4*6	bslbf
valid_period	4*8	bslbf
maker_id	16	uimsbf
maker_code	16	uimsbf
reserved	11	bslbf
playback_control_flag	1	bslbf
write_protect_flag	1	bslbf
is_played_flag	1	bslbf
archive	2	bslbf
ref_thumbnail_index	16	uimsbf
reserved_for_future_use	256	bslbf
}		

FIG.27

write_protect_flag	MEANING
0b	THE PlayList CAN BE ERASED FREELY
1b	THE PlayList CONTENTS SHOULD NOT BE ERASED NOR CHANGED EXCEPT write-protect-flag

FIG.28A

is_played_flag	MEANING
0b	THE PlayList HAS NOT BEEN REPRODUCED SINCE ITS RECORDING
1b	THE PlayList WAS ONCE REPRODUCED SINCE ITS RECORDING

FIG.28B

archive	MEANING
00b	NO MEANING DEFINED
01b	ORIGINAL
10b	COPY
11b	reserved

FIG.28C

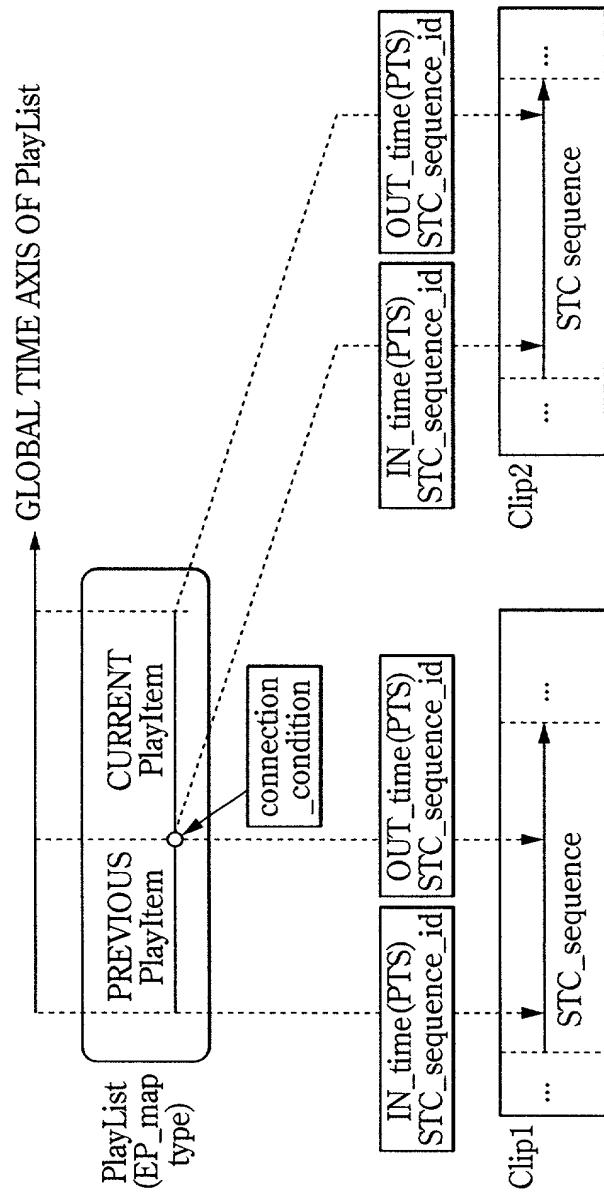


FIG.29

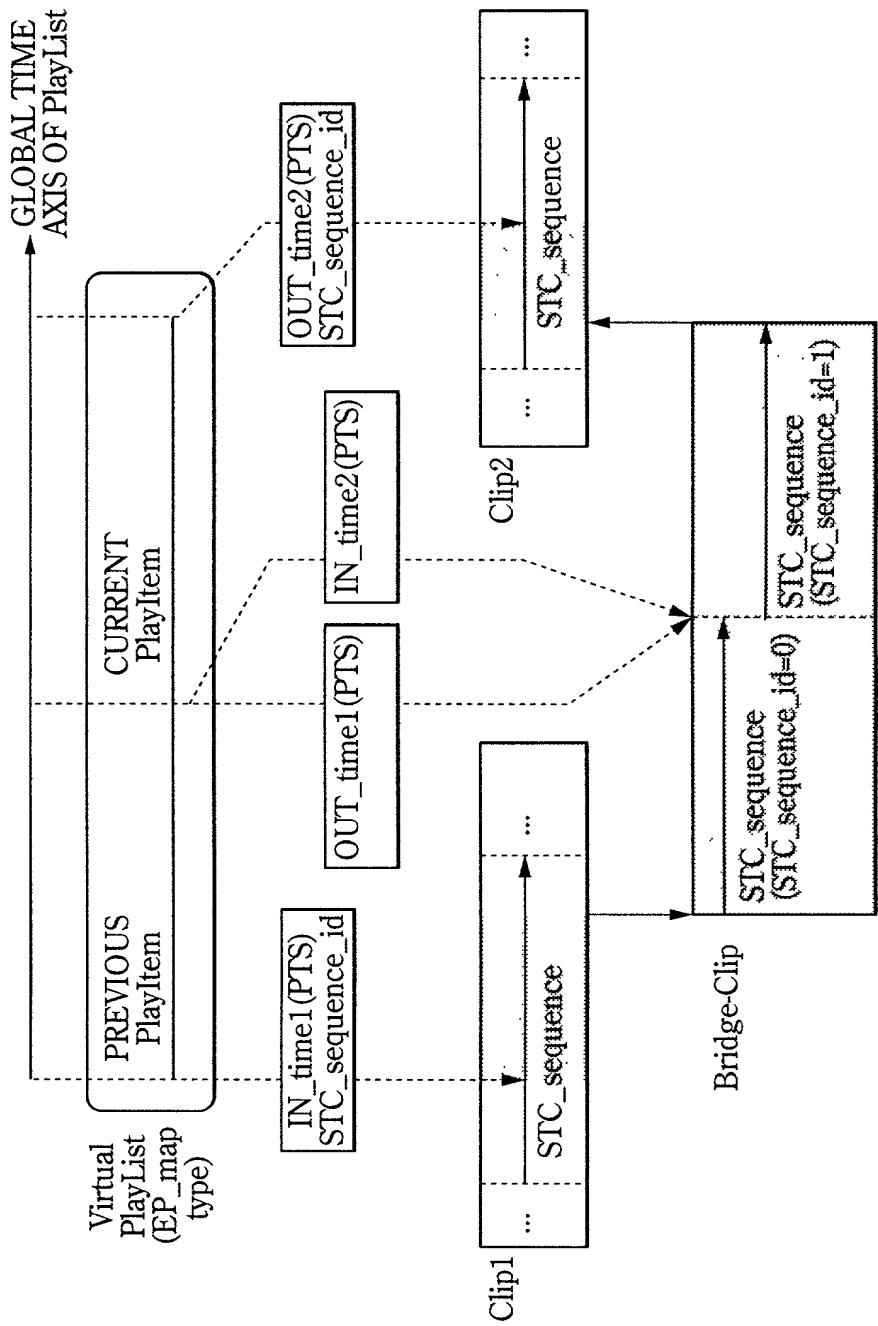


FIG.30

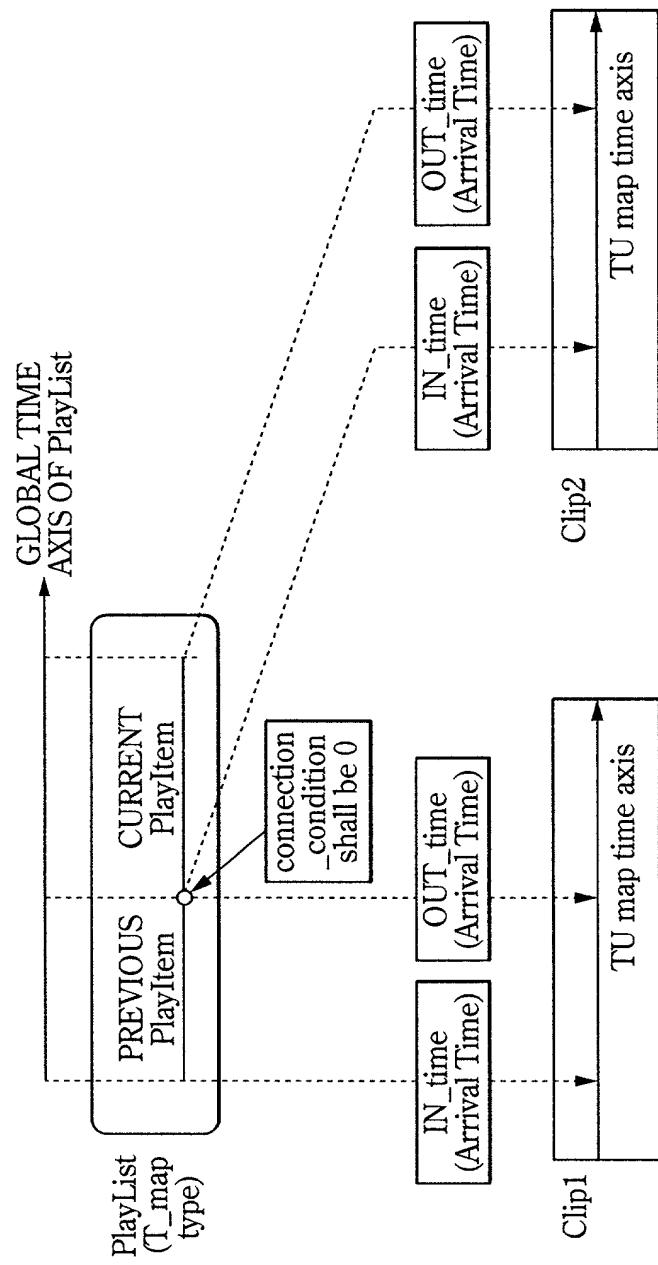


FIG. 3-1

30/100

SYNTAX	NUMBER OF BYTES	ABBREVIATION
PlayItem()		
Clip_information_file_name	8*10	bslbf
reserved	24	bslbf
STC_sequence_id	8	uimsbf
IN_time	32	uimsbf
OUT_time	32	uimsbf
reserved	14	bslbf
connection_condition	2	bslbf
if (<Virtual PlayList>){		
if (<i>connection_condition</i> =='10') {		
BridgeSequenceInfo()		
}		
}		
}		

FIG.32

CPI_type in the PlayList()	SEMANTICS OF IN_time
EP_map type	IN_time MUST INDICATE UPPER 32 BITS OF 33 BIT LENGTH CORRESPONDING TO FIRST PRESENTATION UNIT IN PlayItem
TU_map type	IN_time MUST BE TIME ON TU_map_time_axis, AND MUST BE ROUNDED TO time_unit PRECISION. IN-time IS CALCULATED BY FOLLOWING EQUATION: $\text{IN_time} = \text{TU_start_time} \% 2^{32}$

FIG.33

CPI_type in the PlayList()	SEMANTICS OF OUT_time
EP_map type	<p>OUT_time MUST INDICATE UPPER 32 BITS OF THE VALUE OF Presentation_end_TS CALCULATED BY FOLLOWING EQUATION:</p> $\text{Presentation_end_TS} = \text{PTS_out} + \text{AU_duration}$ <p>WHERE PTS_out IS 33-BIT LONG PTS CORRESPONDING TO LAST PRESENTATION UNIT IN PlayItem. AU_duration IS 90 kHz-DISPLAY TIME OF LAST PRESENTATION UNIT.</p>
TU_map type	<p>OUT_time MUST BE TIME ON TU_map_time_axis AND BE ROUNDED TO time_unit PRECISION. OUT_time IS CALCULATED BY FOLLOWING EQUATION:</p> $\text{OUT_time} = \text{TU_start_time} \% 2^{32}$

FIG.34

connection_condition	MEANING
00	<ul style="list-style-type: none">· CONNECTION OF PREVIOUS PlayItem TO CURRENT PlayItem IS NOT SURE AS TO SEAMLESS REPLAY.· IF CPI_type OF PlayList IS TU_map type, THIS VALUE MUST BE SET IN connection_condition.
01	<ul style="list-style-type: none">· THIS STATE IS ALLOWED ONLY WHEN CPI_type OF PlayList IS EP_map type.· PREVIOUS PlayItem AND CURRENT PlayItem INDICATE DIVISION BECAUSE OF NON-CONTINUOUS POINT OF SYSTEM TIMEBASE (STC BASE).
10	<ul style="list-style-type: none">· THIS STATE IS ALLOWED ONLY WHEN CPI_type OF PlayList IS EP_map type.· THIS STATE IS ALLOWED ONLY FOR Virtual PlayList.· CONNECTION OF PREVIOUS PlayItem TO CURRENT PlayItem IS SURE AS TO SEAMLESS REPLAY.· PREVIOUS PlayItem IS CONNECTED TO CURRENT PlayItem USING BridgeSequence. DVR MPEG-2 TRANSPORT STREAM MUST OBEY DVR-STD AS LATER DESCRIBED.
11	<ul style="list-style-type: none">· THIS STATE IS ALLOWED ONLY WHEN CPI_type OF PlayList IS EP_map type.· CONNECTION OF PREVIOUS PlayItem TO CURRENT Play Item IS SURE AS TO SEAMLESS REPLAY.· PREVIOUS PlayItem IS CONNECTED TO CURRENT PlayItem WITHOUT USING BridgeSequence. DVR MPEG-2 TRANSPORT STREAM MUST OBEY DVR-STD AS LATER DESCRIBED.

FIG.35

34/100

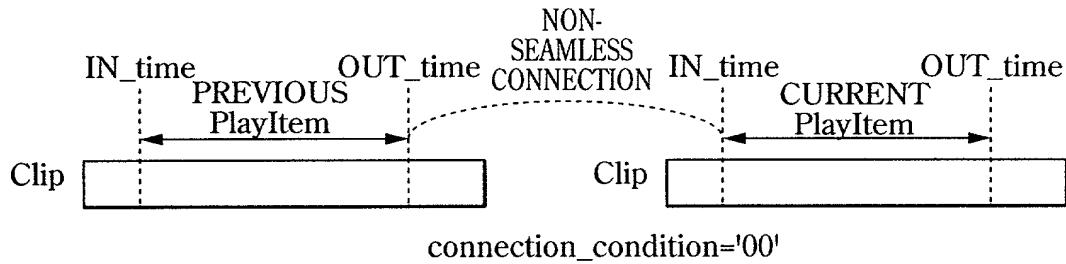


FIG.36A

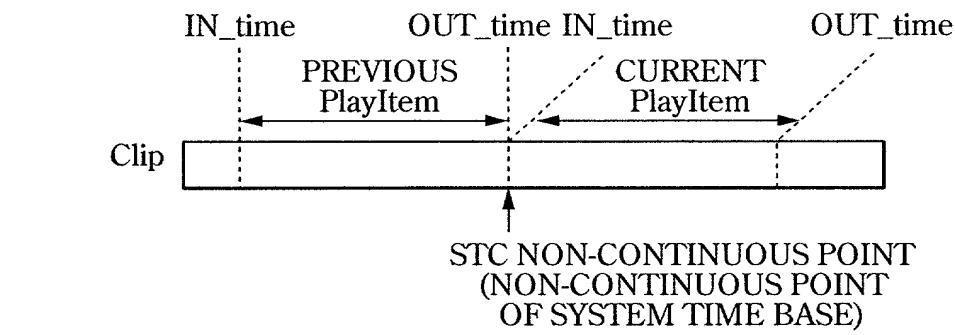


FIG.36B

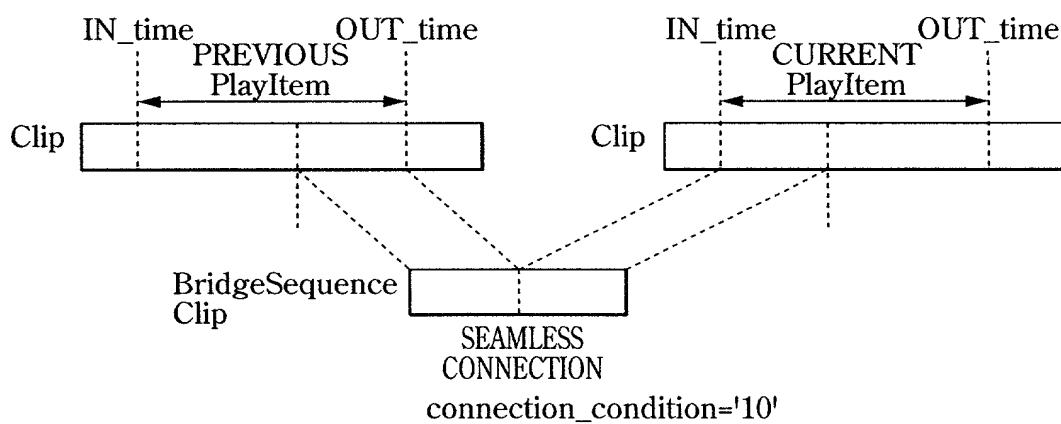


FIG.36C

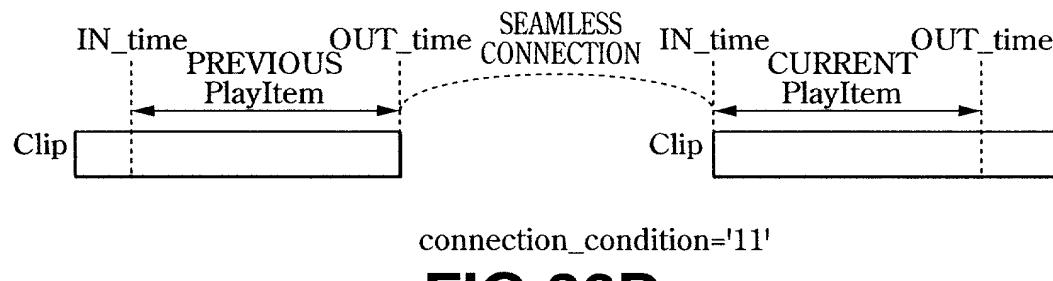
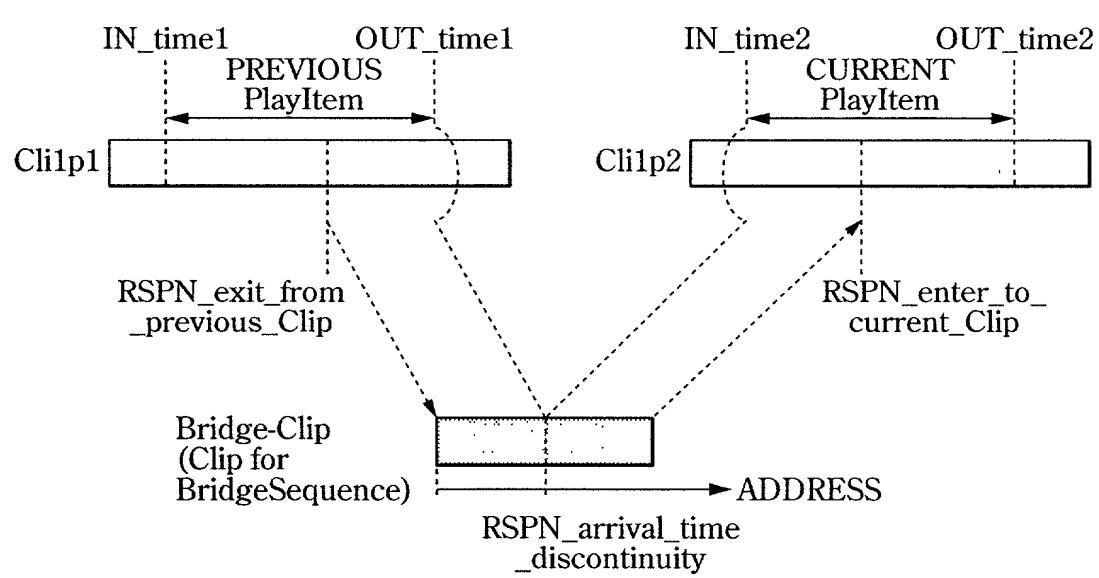


FIG.36D

**FIG.37**

SYNTAX	NUMBER OF BYTES	ABBREVIATION
BridgeSequenceInfo()		
Bridge_Clip_information_file_name	8*10	bslbf
RSPN_exit_from_previous_Clip	32	uimsbf
RSPN_enter_to_current_Clip	32	uimsbf
}		

FIG.38

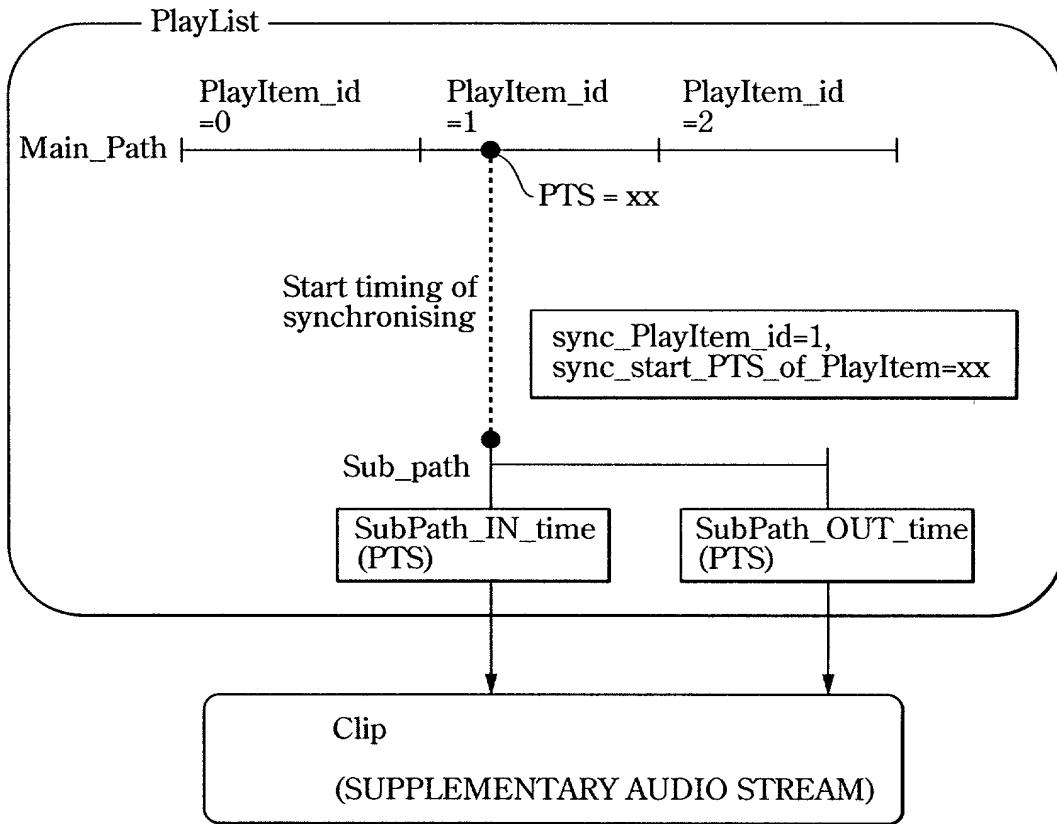


FIG.39

SYNTAX	NUMBER OF BYTES	ABBREVIATION
SubPlayItem(){		
Clip_Information_file_name	8*10	bslbf
SubPath_type	8	bslbf
sync_PlayItem_id	8	uimsbf
sync_start PTS of PlayItem	32	uimsbf
SubPath_IN_time	32	uimsbf
SubPath_OUT_time	32	uimsbf
}		

FIG.40

SubPath_type	MEANING
0x00	Auxiliary audio steam path
0x01-0xff	reserved

FIG.41

SYNTAX	NUMBER OF BYTES	ABBREVIATION
PlayListMark(){		
version_number	8*4	bslbf
length	32	uimsbf
number_of_PlayList_marks	16	uimsbf
for (i=0;i< <i>number_of_PlayList_marks</i> ;i++){		
reserved	8	bslbf
mark_type	8	bslbf
mark_time_stamp	32	uimsbf
PlayItem_id	8	uimsbf
reserved	24	uimsbf
character_set	8	bslbf
name_length	8	uimsbf
mark_name	8*256	bslbf
ref_thumbnail_index	16	uimsbf
}		
}		

FIG.42

Mark_type	MEANING	COMMENT
0x00	resume-mark	REPLAY RESUME POINT. THE NUMBER OF REPLAY RESURE POINTS DEFINED IN PlayListMark() MUST BE 0 OR 1.
0x01	book-mark	REPLAY ENTRY POINT OF PlayList. THIS MARK CAN BE SET BY USER AND USED AS MARK SPECIFYING START POINT OF FAVORITE SCENE.
0x02	skip-mark	SKIP MARK POINT. PLAYER SKIPS PROGRAM FROM THIS POINT TO THE END OF PROGRAM. THE NUMBER OF SKIP MARK POINTS DEFINED IN PlayListMark() MUST BE 0 OR 1.
0x03-0x8F	reserved	
0x90-0xFF	reserved	Reserved for ClipMark()

FIG.43

CPI_type in the PlayList0	SEMANTICS OF mark_time_stamp
EP_map type	mark_time_stamp MUST INDICATE UPPER 32 BITS OF 33 BIT LENGTH PTS CORRESPONDING TO PRESENTATION UNIT REFERENCED BY MARK.
TU_map type	mark_time_stamp MUST BE TIME ON TU_map_time_axis AND MUST BE ROUNDED TO time_unit PRECISION. mark_time_stamp IS CALCULATED BY FOLLOWING EQUATION: $\text{mark_time_stamp} = \text{TU_start_time \%}2^{32}$

FIG.44

SYNTAX	NUMBER OF BYTES	ABBREVIATION
<code>zzzzz.clpi {</code>		
<code>STC_Info_Start_address</code>	32	uimsbf
<code>ProgramInfo_Start_address</code>	32	uimsbf
<code>CPI_Start_address</code>	32	uimsbf
<code>ClipMark_Start_address</code>	32	uimsbf
<code>MakersPrivateData_Start_address</code>	32	uimsbf
<code>reserved</code>	96	bslbf
<code>ClipInfo()</code>		
<code>for (i=0;i<N1;i++){</code>		
<code>padding_word</code>	16	bslbf
<code>}</code>		
<code>STC_Info()</code>		
<code>for (i=0;i<N2;i++){</code>		
<code>padding_word</code>	16	bslbf
<code>}</code>		
<code>ProgramInfo()</code>		
<code>for (i=0;i<N3;i++){</code>		
<code>padding_word</code>	16	bslbf
<code>}</code>		
<code>CPI()</code>		
<code>for (i=0;i<N4;i++){</code>		
<code>padding_word</code>	16	bslbf
<code>}</code>		
<code>ClipMark()</code>		
<code>for (i=0;i<N5;i++){</code>		
<code>padding_word</code>	16	bslbf
<code>}</code>		
<code>MakersPrivateData()</code>		
<code>}</code>		

FIG.45

SYNTAX	NUMBER OF BYTES	ABBREVIATION
ClipInfo()		
version_number	8*4	bslbf
length	32	uimsbf
Clip_stream_type	8	bslbf
offset_SPN	32	uimsbf
TS_recording_rate	24	uimsbf
reserved	8	bslbf
record_time_and_date	4*14	bslbf
reserved	8	bslbf
duration	4*6	bslbf
reserved	7	bslbf
time_controlled_flag	1	bslbf
TS_average_rate	24	uimsbf
<i>if(Clip_stream_type==1) // Bridge-Clip AV stream</i>		
RSPN_arrival_time_discontinuity	32	uimsbf
<i>else</i>		
reserved	32	bslbf
reserved_for_system_use	144	bslbf
reserved	11	bslbf
is_format_identifier_valid	1	bslbf
is_original_network_ID_valid	1	bslbf
is_transport_stream_ID_valid	1	bslbf
is_service_ID_valid	1	bslbf
is_country_code_valid	1	bslbf
format_identifier	32	bslbf
original_network_ID	16	uimsbf
transport_stream_ID	16	uimsbf
service_ID	16	uimsbf
country_code	24	bslbf
stream_format_name	16*8	bslbf
reserved_for_fortune_use	256	bslbf
}		

FIG.46

Clip_stream_type	MEANING
0	Clip AV STREAM
1	Bridge-Clip AV STREAM
2-255	Reserved

FIG.47

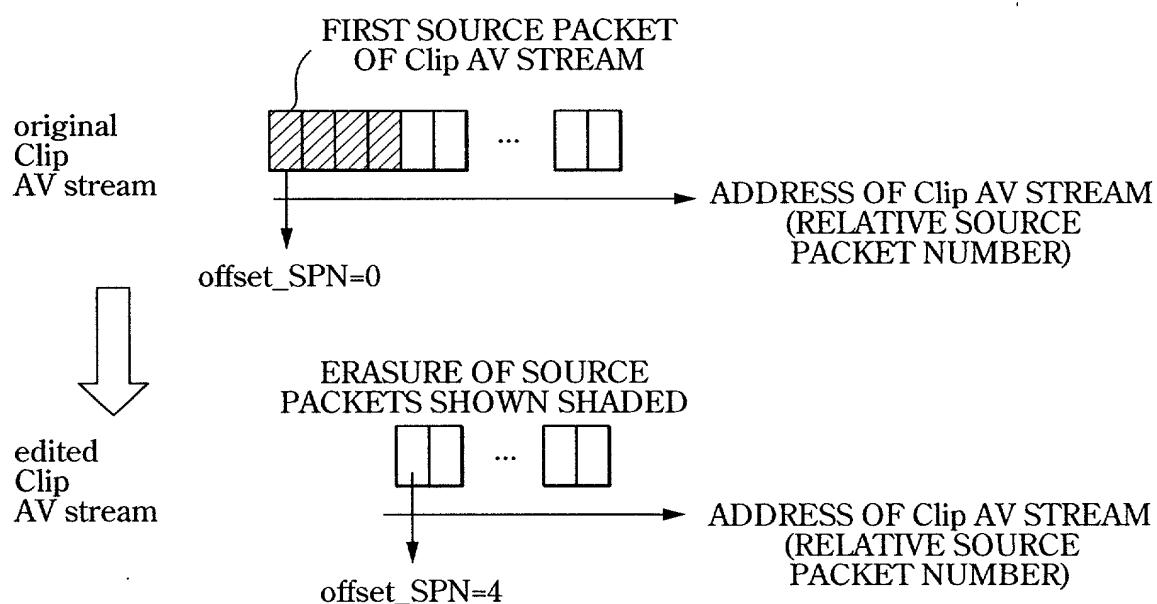


FIG.48

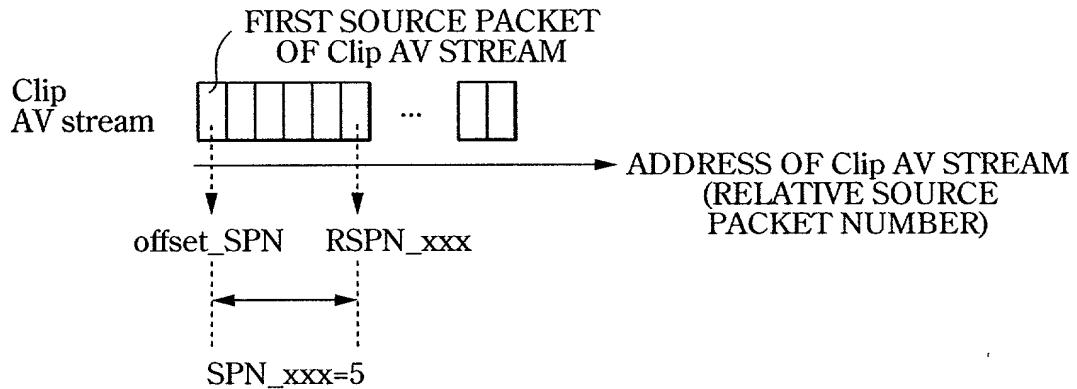


FIG.49

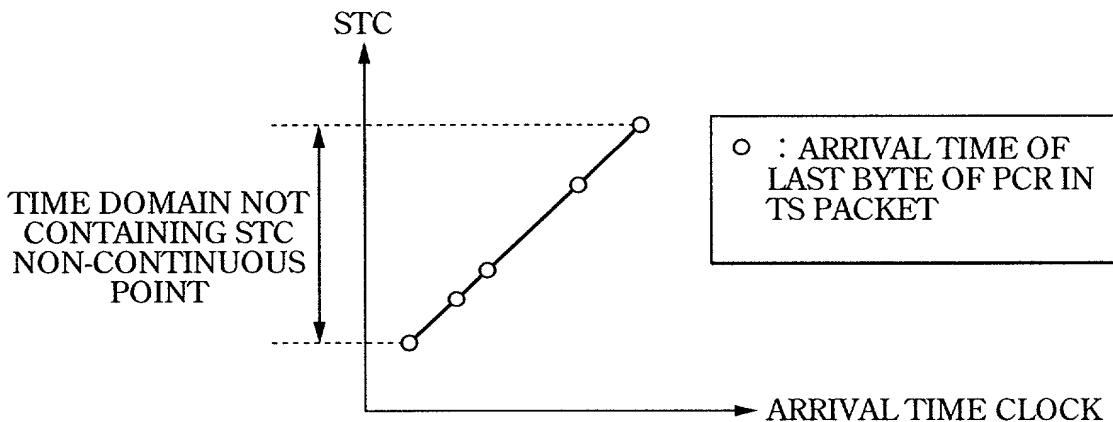


FIG.50A

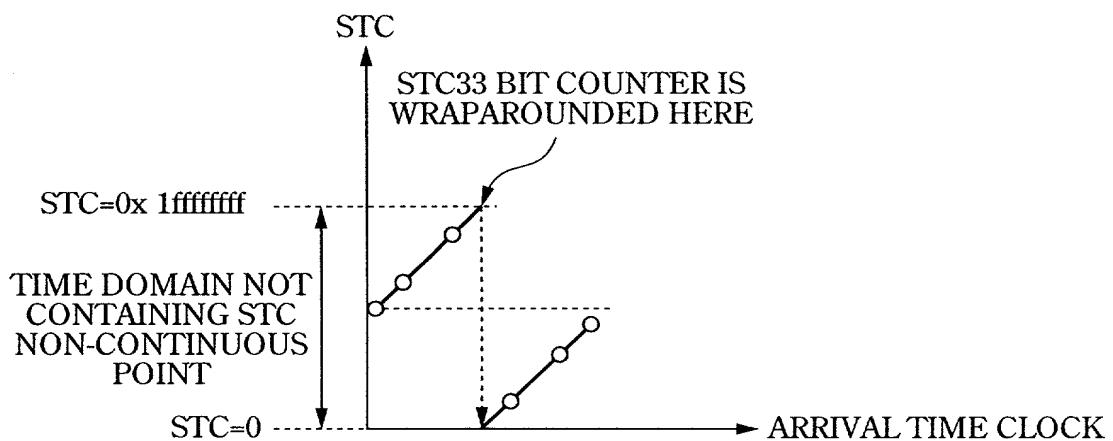


FIG.50B

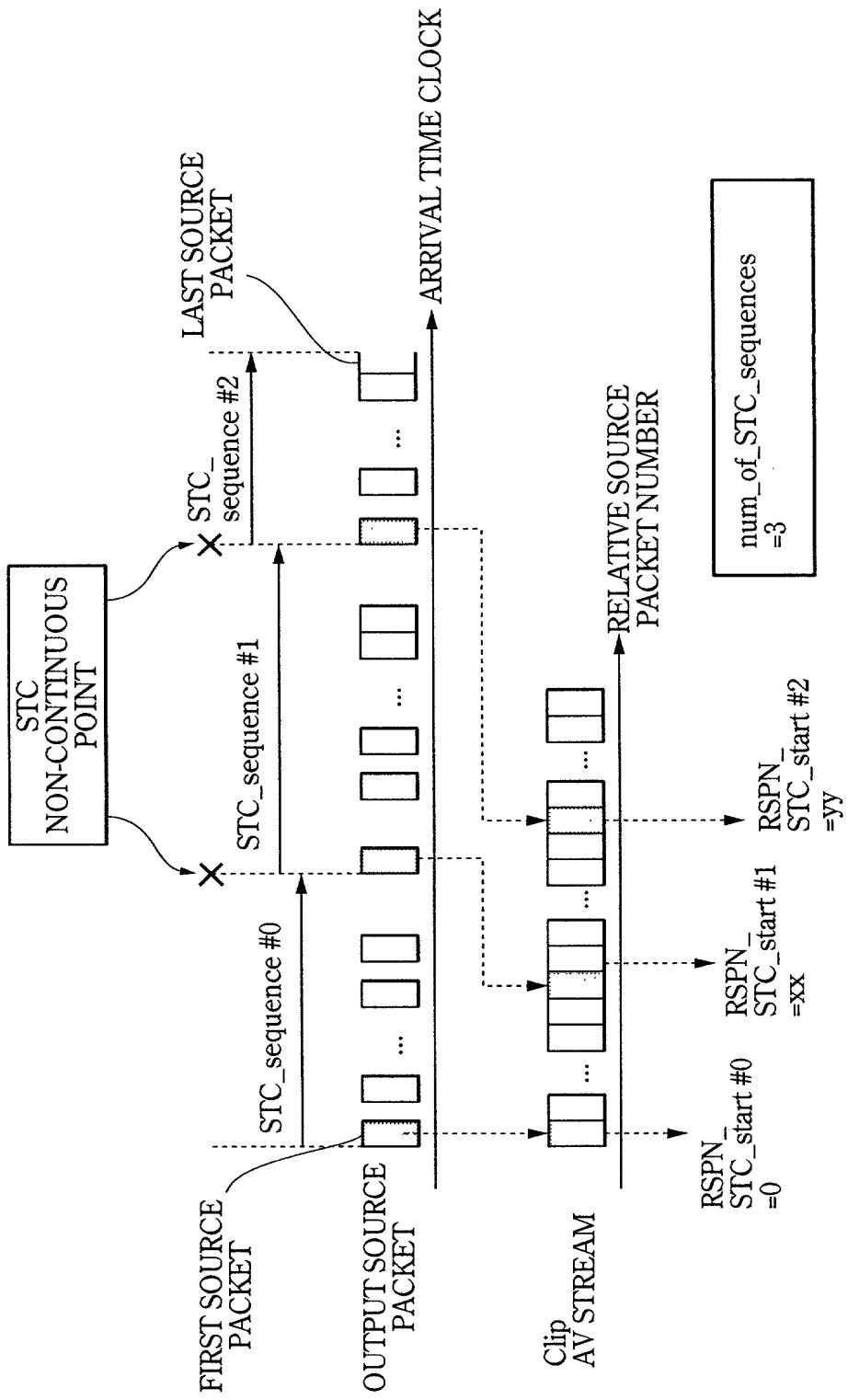


FIG.51

SYNTAX	NUMBER OF BYTES	ABBREVIATION
STC_Info(){		
version_number	8*4	bslbf
length	32	uimsbf
if (length !=0){		
reserved	8	bslbf
num_of_STC_sequences	8	uimsbf
for (STC_sequence_id=0;		
STC_sequence_id<num_of_STC_sequences;		
STC_sequence_id++){		
resereved	32	bslbf
RSPN_STC_start	32	uimsbf
}		
}		
}		

FIG.52

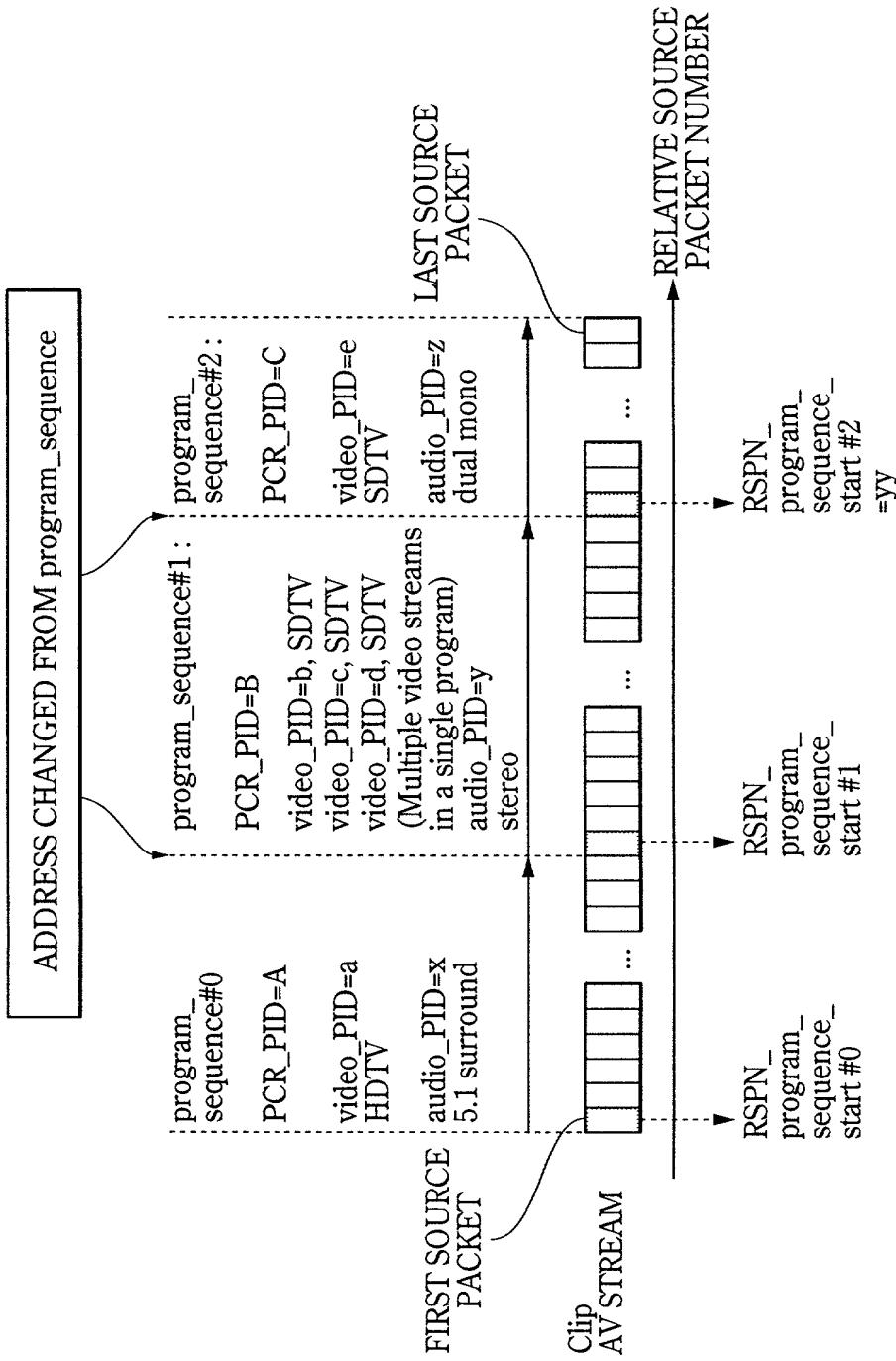


FIG.53

SYNTAX	NUMBER OF BYTES	ABBREVIATION
ProgramInfo()		
version_number	8*4	bslbf
length	32	uimsbf
if (length !=0){		
reserved	8	bslbf
number_of_program_sequences	8	uimsbf
for (i=0;i< <i>number_of_program_sequences</i> ;i++){		
RSPN_program_sequence_start	32	uimsbf
reserved	48	bslbf
PCR_PID	16	bslbf
number_of_videos	8	uimsbf
number_of_audios	8	uimsbf
for (k=0;k< <i>number_of_videos</i> ;k++){		
video_stream_PID	16	bslbf
VideoCodingInfo()		
}		
for (k=0;k< <i>number_of_audios</i> ;k++){		
audio_stream_PID	16	bslbf
AudioCodingInfo()		
}		
}		
}		
}		

FIG.54

SYNTAX	NUMBER OF BYTES	ABBREVIATION
VideoCodingInfo()		
video_format	8	uimsbf
frame_rate	8	uimsbf
display_aspect_ratio	8	uimsbf
reserved	8	bslbf
}		

FIG.55

video_format	MEANING
0	480i
1	576i
2	480p (including 640×480p format)
3	1080i
4	720p
5	1080p
6-254	reserved
255	No information

FIG.56

frame_rate	MEANING
0	forbidden
1	24 000/1001 (23.976...)
2	24
3	25
4	30 000/1001 (29.97..)
5	30
6	50
7	60 000/1001 (59.94..)
8	60
9-254	reserved
255	No information

FIG.57

display_aspect_ratio	MEANING
0	forbidden
1	reserved
2	4:3 display aspect ratio
3	16:9 display aspect ratio
4-254	reserved
255	No information

FIG.58

SYNTAX	NUMBER OF BYTES	ABBREVIATION
AudioCodingInfo()		
audio_format	8	uimsbf
audio_component_type	8	uimsbf
sampling_frequency	8	uimsbf
reserved	8	bslbf
}		

FIG.59

audio_coding	MEANING
0	MPEG-1 audio layer I or II
1	Dolby AC-3 audio
2	MPEG-2 AAC
3	MPEG-2 multi-channel audio, backward compatible to MPEG-1
4	SESF LPCM audio
5-254	reserved
255	No information

FIG.60

audio_component_type	MEANING
0	single mono channel
1	dual mono channel
2	stereo (2-channel)
3	multi-lingual, multi-channel
4	surround sound
5	audio description for the visually impaired
6	audio for the hard of hearing
7-254	reserved
255	No information

FIG.61

sampling_frequency	MEANING
0	48 kHz
1	44.1 kHz
2	32 kHz
3-254	reserved
255	No information

FIG.62

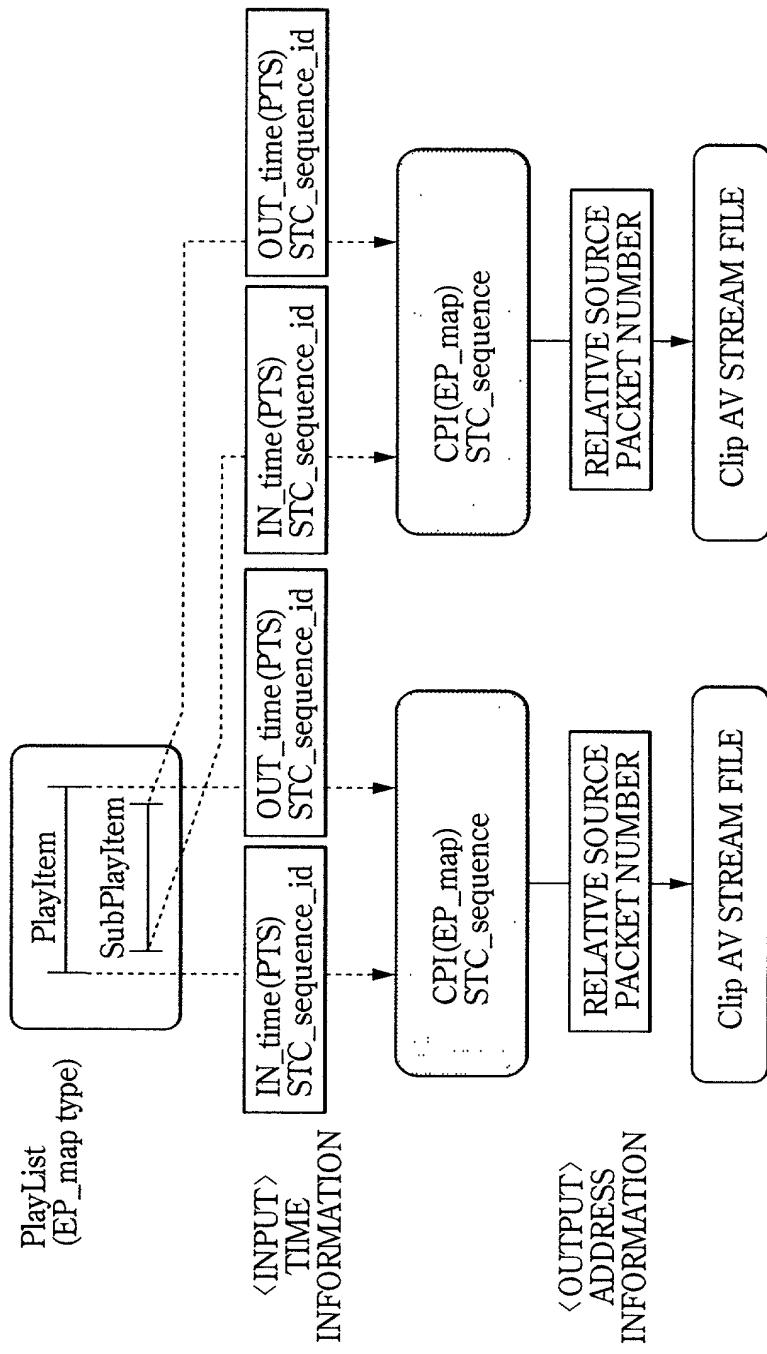


FIG. 63

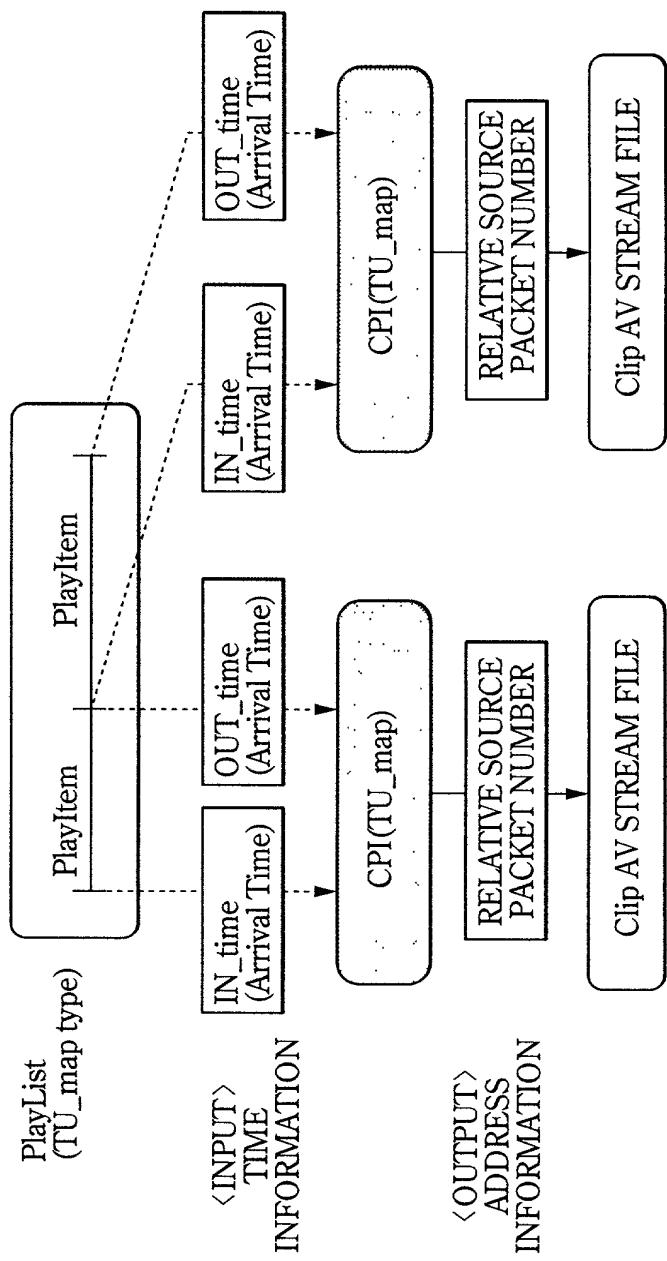


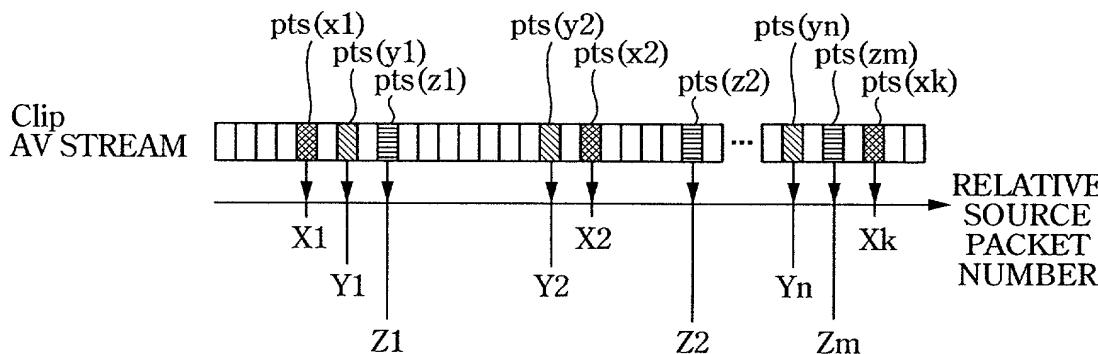
FIG.64

SYNTAX	NUMBER OF BYTES	ABBREVIATION
CPI()		
version_number	8*4	bslbf
length	32	uimsbf
reserved	15	bslbf
CPI_type	1	bslbf
if (CPI_type==0)		
EP_map()		
else		
TU_map()		
}		

FIG.65

CPI_type	MEANING
0	EP map type
1	TU map type

FIG.66



- : SOURCE PACKET CONTAINING FIRST BYTE OF SEQUENCE HEADER video_PID=x
- : SOURCE PACKET CONTAINING FIRST BYTE OF SEQUENCE HEADER video_PID=y
- : SOURCE PACKET CONTAINING FIRST BYTE OF SEQUENCE HEADER video_PID=z

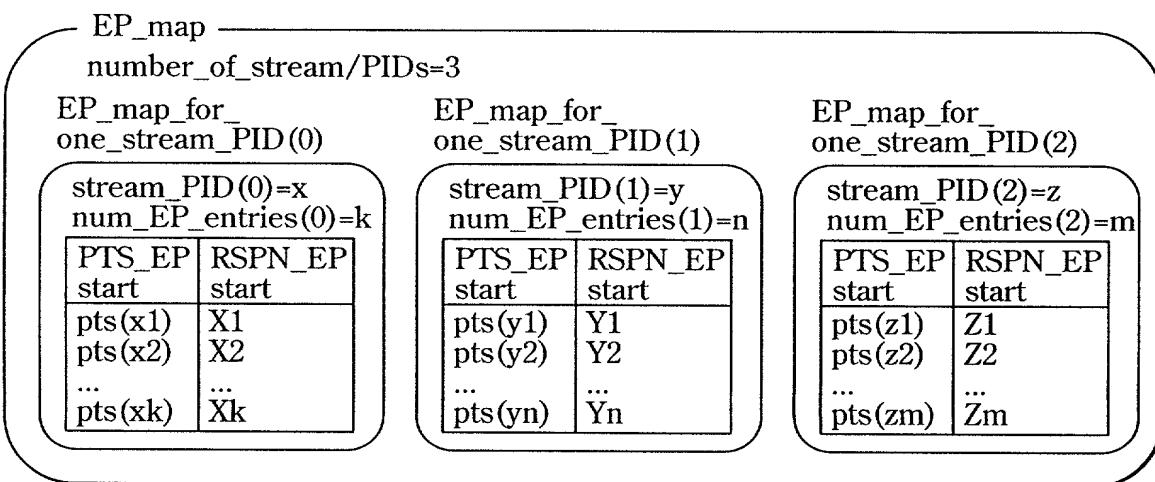


FIG.67

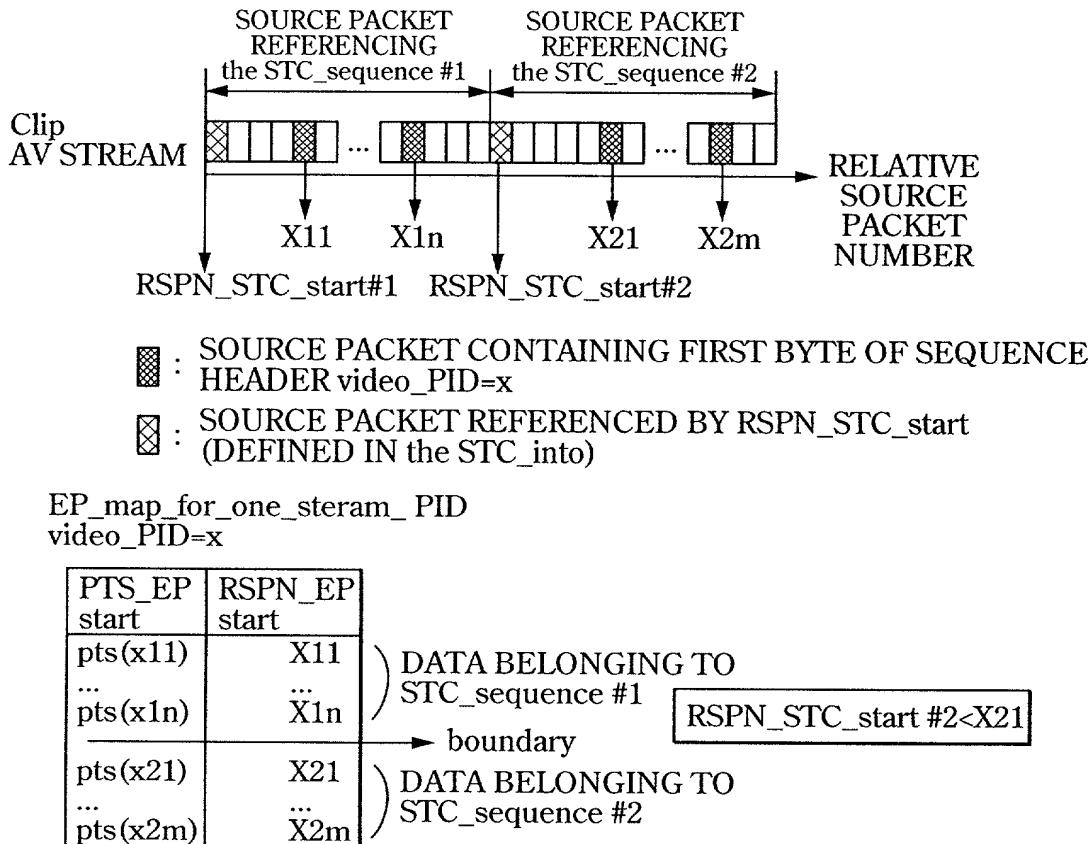


FIG.68

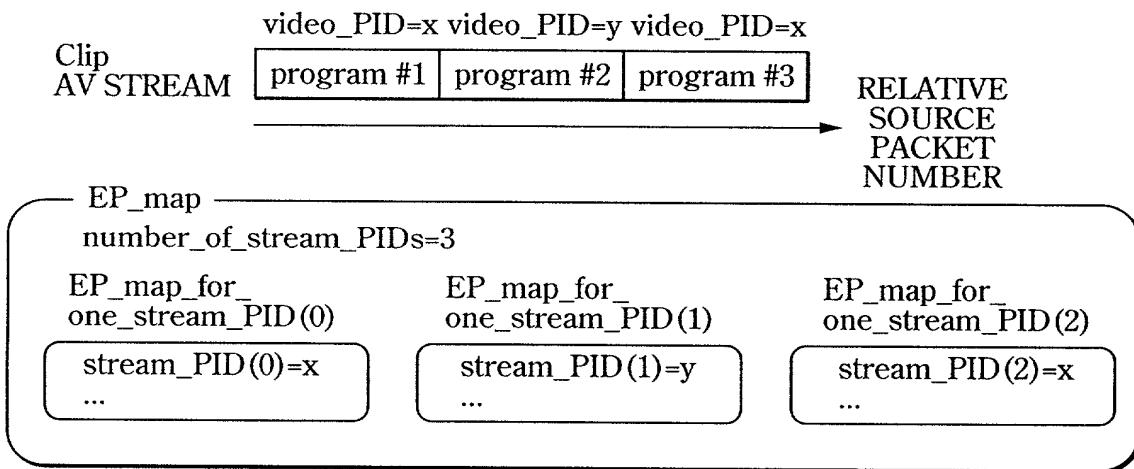


FIG.69

SYNTAX	NUMBER OF BYTES	ABBREVIATION
EP_map0{		
reserved	12	bslbf
EP_type	4	uimsbf
number_of_stream_PIDs	16	uimsbf
for (k=0;k< <i>number_of_stream_PIDs</i> ;k++){		
stream_PID(k)	16	bslbf
num_EP_entries(k)	32	uimsbf
EP_map_for_one_stream_PID_Start_address(k)	32	uimsbf
}		
for (i=0;i<X;i++){		
padding_word	16	bslbf
}		
for (k=0;k< <i>number_of_stream_PIDs</i> ;k++){		
EP_map_for_one_stream_PID(num_EP_entries(k))		
for (i=0;i<Y;i++){		
padding_word	16	bslbf
}		
}		
}		

FIG.70

EP_type	MEANING
0	video
1	audio
2-15	reserved

FIG.71

SYNTAX	NUMBER OF BYTES	ABBREVIATION
EP_map_for_one_stream_PID(N) {		
for (i=0;i< N ;i++) {		
PTS_EP_start	32	uimsbf
RSPN_EP_start	32	uimsbf
}		
}		

FIG.72

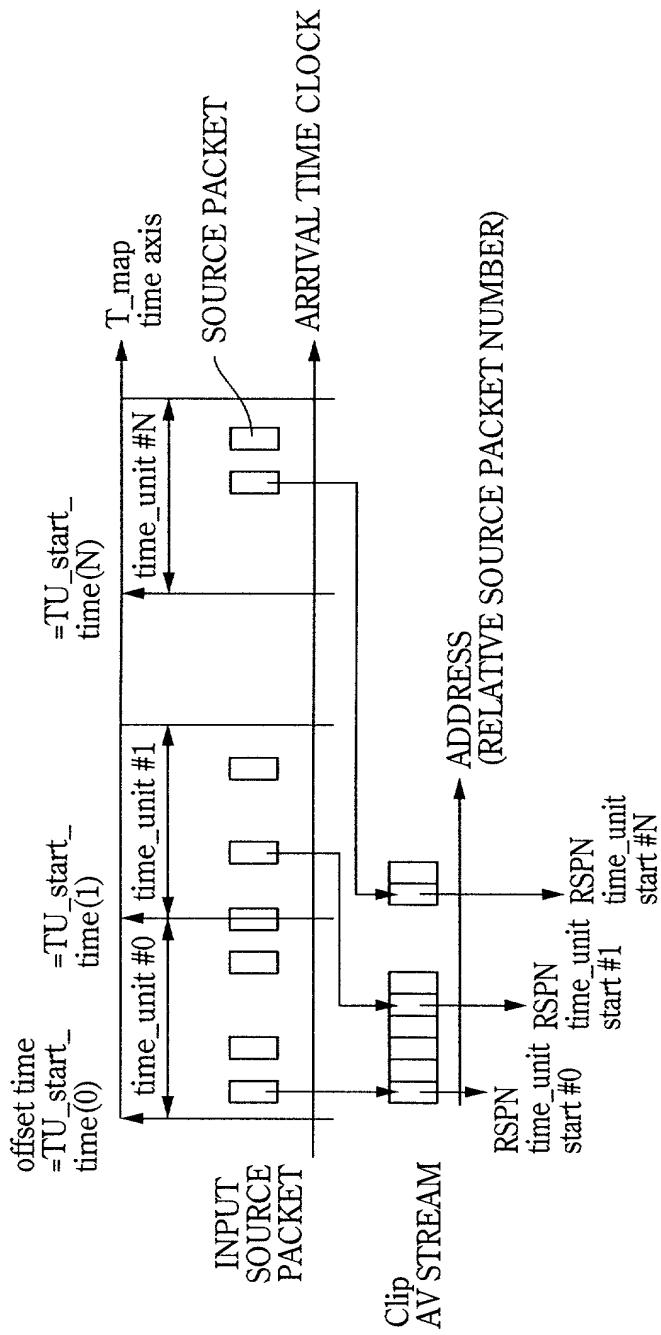


FIG.73

SYNTAX	NUMBER OF BYTES	ABBREVIATION
TU_map()		
offset_time	32	bslbf
time_unit_size	32	uimsbf
number_of_time_unit_entries	32	uimsbf
for (k=0;k<number_of_time_unit_entries;k++)		
RSPN_time_unit_start	32	uimsbf
}		

FIG.74

SYNTAX	NUMBER OF BYTES	ABBREVIATION
ClipMark0{		
version_number	8*4	bslbf
length	32	uimsbf
number_of_Clip_marks	16	uimsbf
for (i=0; i< number_of_clip_marks ; i++){		
reserved	8	bslbf
mark_type	8	bslbf
mark_time_stamp	32	uimsbf
STC_sequence_id	8	uimsbf
reserved	24	bslbf
character_set	8	bslbf
name_length	8	uimsbf
mark_name	8*256	bslbf
ref_thumbnail_index	16	uimsbf
}		
}		

FIG.75

Mark_type	MEANING	COMMENT
0x00-0x8F	reserved	Reserved for PlayListMark0
0x90	Event-start mark	MARK POINT INDICATING PROGRAM START POINT
0x91	Local event-start mark	MARK POINT INDICATING LOCAL SCENE IN PROGRAM
0x92	Scene-start mark	MARK POINT SHOWING SCENE CHANGE POINT
0x93-0xFF	reserved	

FIG.76

CPI_type in the PlayList()	SEMANTICS OF mark_time_stamp
EP_map type	mark_time_stamp MUST INDICATE UPPER 32 BITS OF 33 BIT LENGTH PTS CORRESPONDING TO PRESENTATION UNIT REFERENCED BY MARK.
TU_map type	mark_time_stamp MUST BE TIME ON TU_map_time_axis AND MUST BE ROUNDED TO time_unit PRECISION. mark_time_stamp IS CALCULATED BY FOLLOWING EQUATION: $\text{mark_time_stamp} = \text{TU_start_time \%}2^{32}$

FIG.77

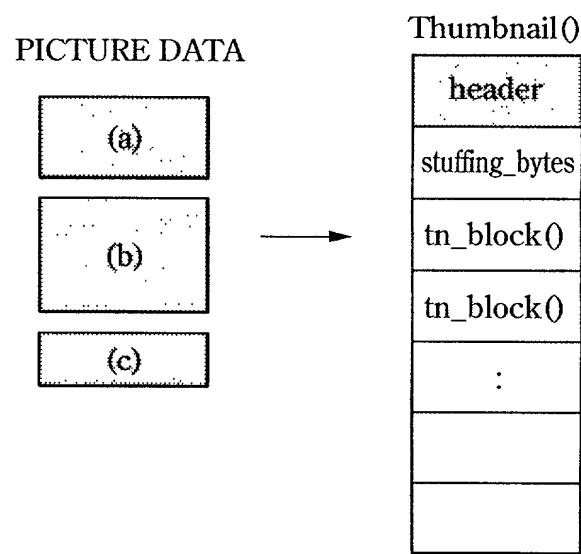
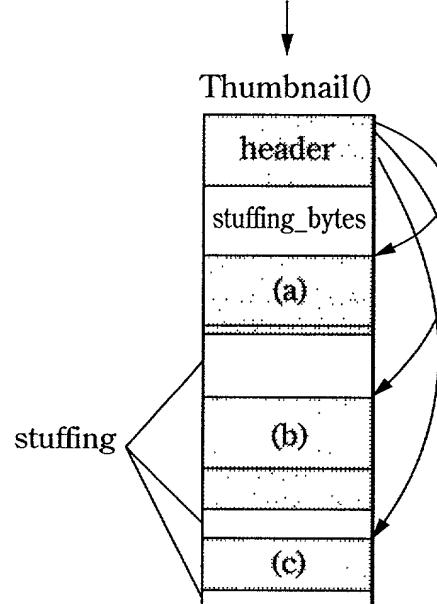
SYNTAX	NUMBER OF BYTES	ABBREVIATION
menu.thmb/mark.thmb {		
reserved	256	bslbf
Thumbnail()		
for (i=0; i<N1; i++)		
padding_word	16	bslbf
}		

FIG.78

SYNTAX	NUMBER OF BYTES	ABBREVIATION
Thumbnail0{		
version_number	8*4	char
length	32	uimsbf
if (length !=0){		
tn_blocks_start_address	32	bslbf
number_of_thumbnails	16	uimsbf
tn_block_size	16	uimsbf
number_of_tn_blocks	16	uimsbf
reserved	16	bslbf
for (i=0; i<number_of_thumbnails; i++){		
thumbnail_index	16	uimsbf
thumbnail_picture_format	8	bslbf
reserved	8	bslbf
picture_data_size	32	uimsbf
start_tn_block_number	16	uimsbf
x_picture_length	16	uimsbf
y_picture_length	16	uimsbf
reserved	16	uimsbf
}		
stuffing_bytes	8*2*L1	bslbf
for(k=0; k<number_of_tn_blocks; k++){		
tn_block	tn_block_size*1024*8	
}		
}		
}		

FIG.79

Thumbnail_picture_format	MEANING
0x00	MPEG-2 Video I-picture
0x01	DCF (restricted JPEG)
0x02	PNG
0x03-0xff	reserved

FIG.80**FIG.81A****FIG.81 B**

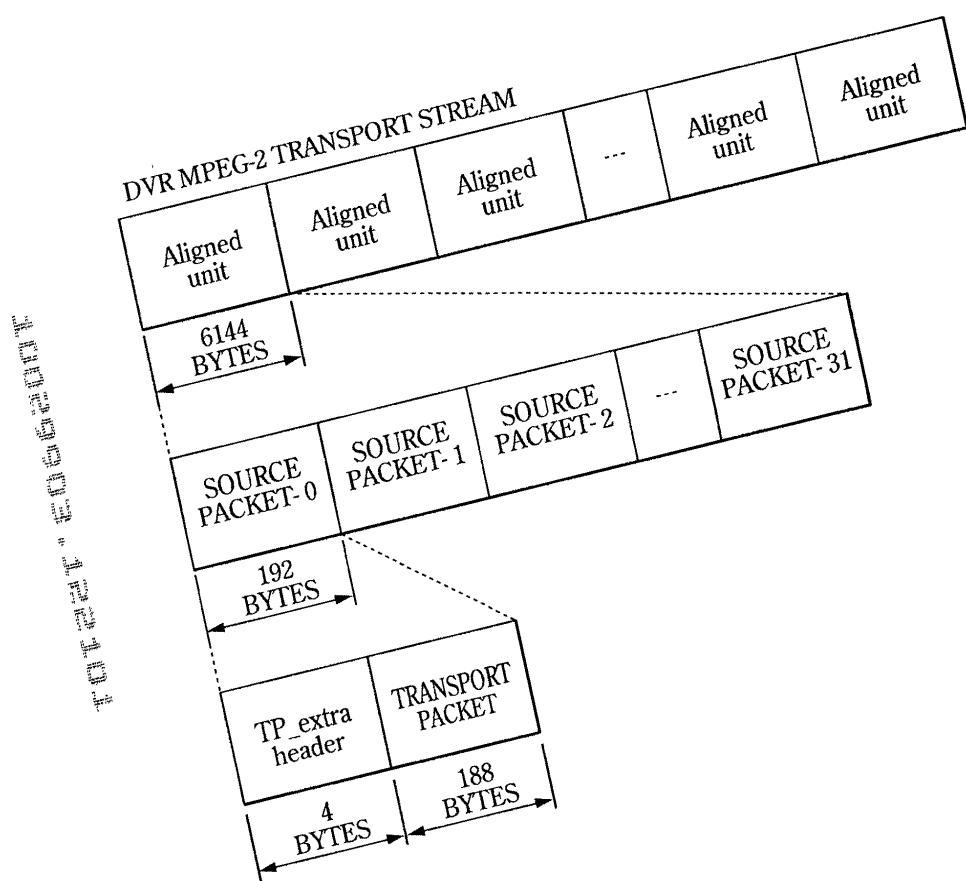


FIG.82

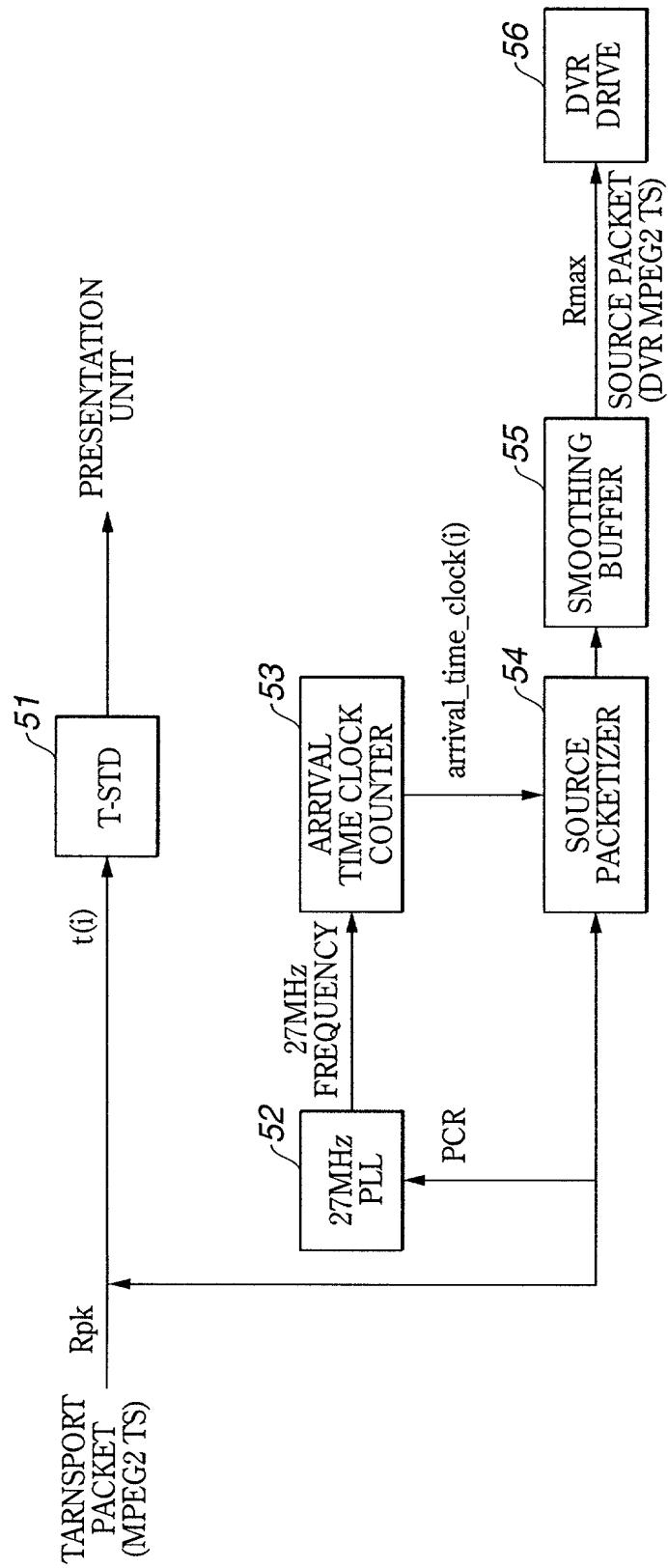


FIG. 83

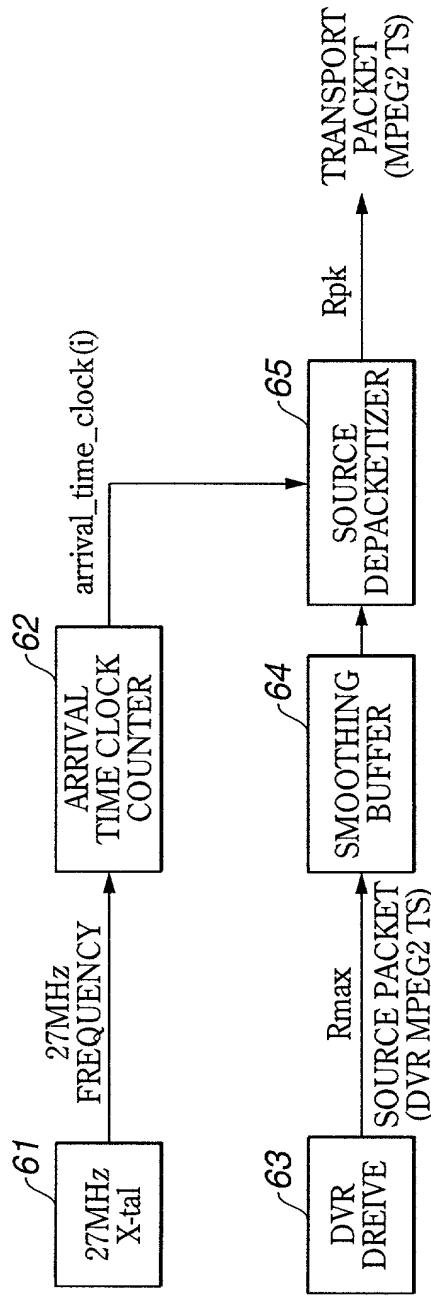


FIG.84

SYNTAX	NUMBER OF BYTES	ABBREVIATION
source_packet()		
TP_extra_header()		
transport_packet()		
}		

FIG.85

SYNTAX	NUMBER OF BYTES	ABBREVIATION
TP_extra_header()		
copy_permission_indicator	2	uimsbf
arrival_time_stamp	30	uimsbf
}		

FIG.86

copy_permission _indicator	MEANING
00	copy free
01	no more copy
10	copy once
11	copy prohibited

FIG.87

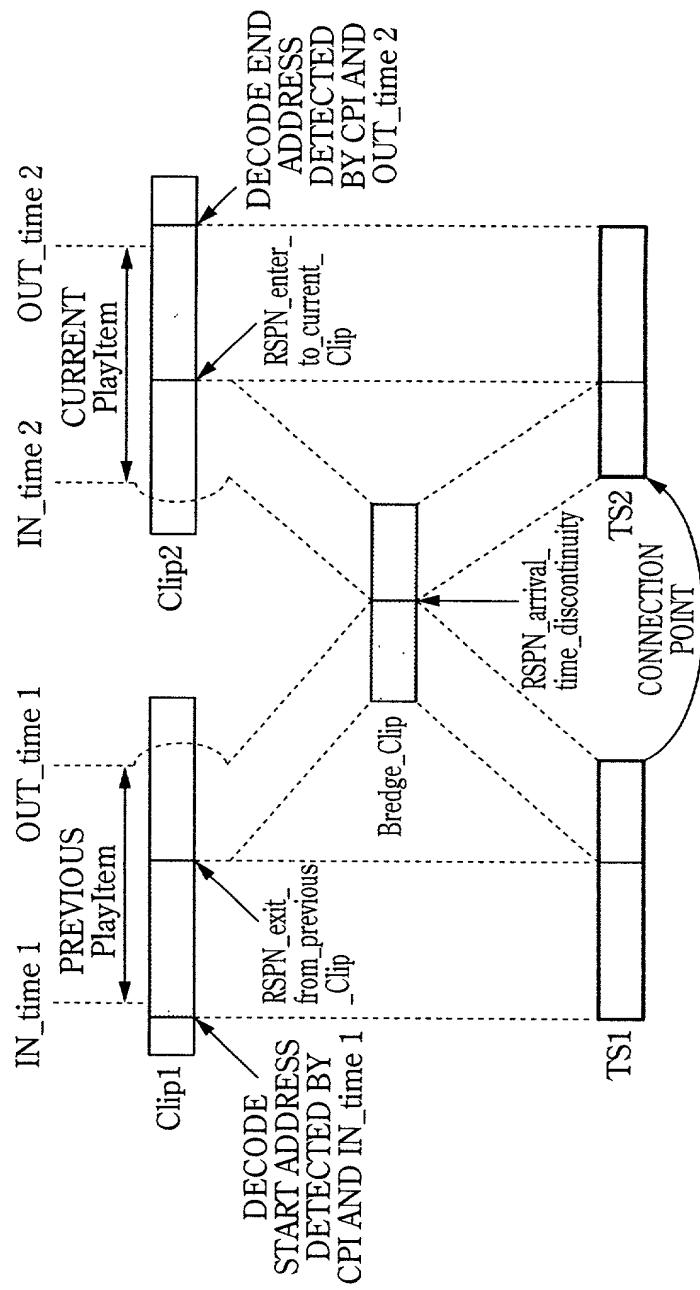


FIG.88

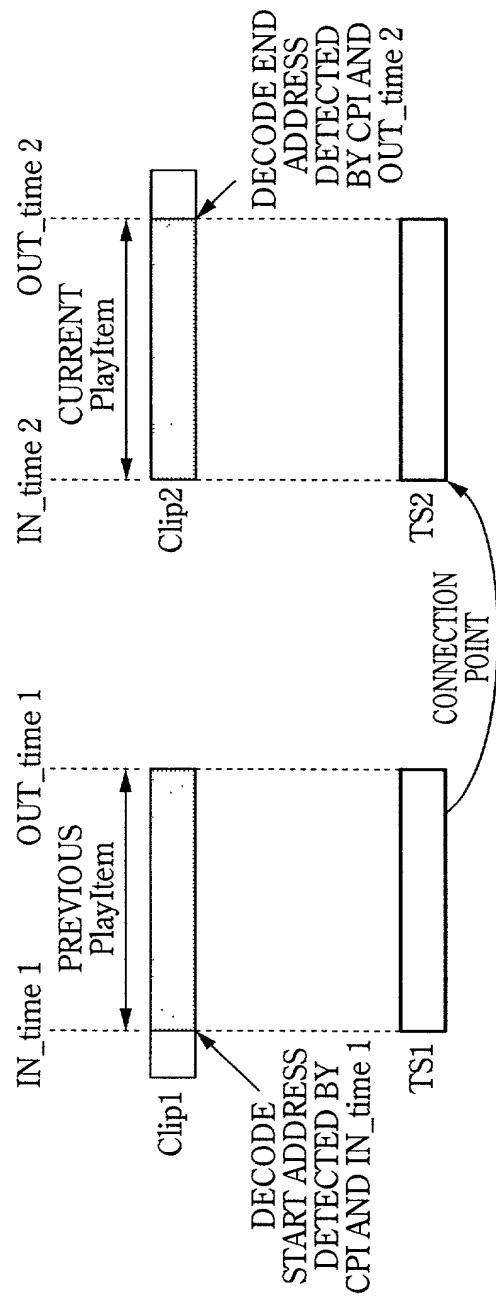
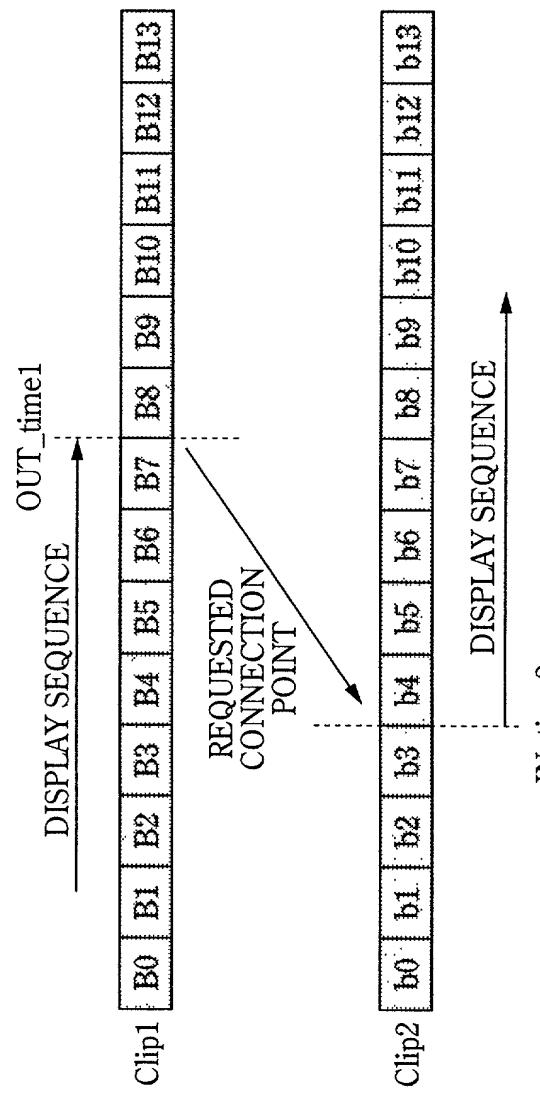


FIG.89

**FIG.90**

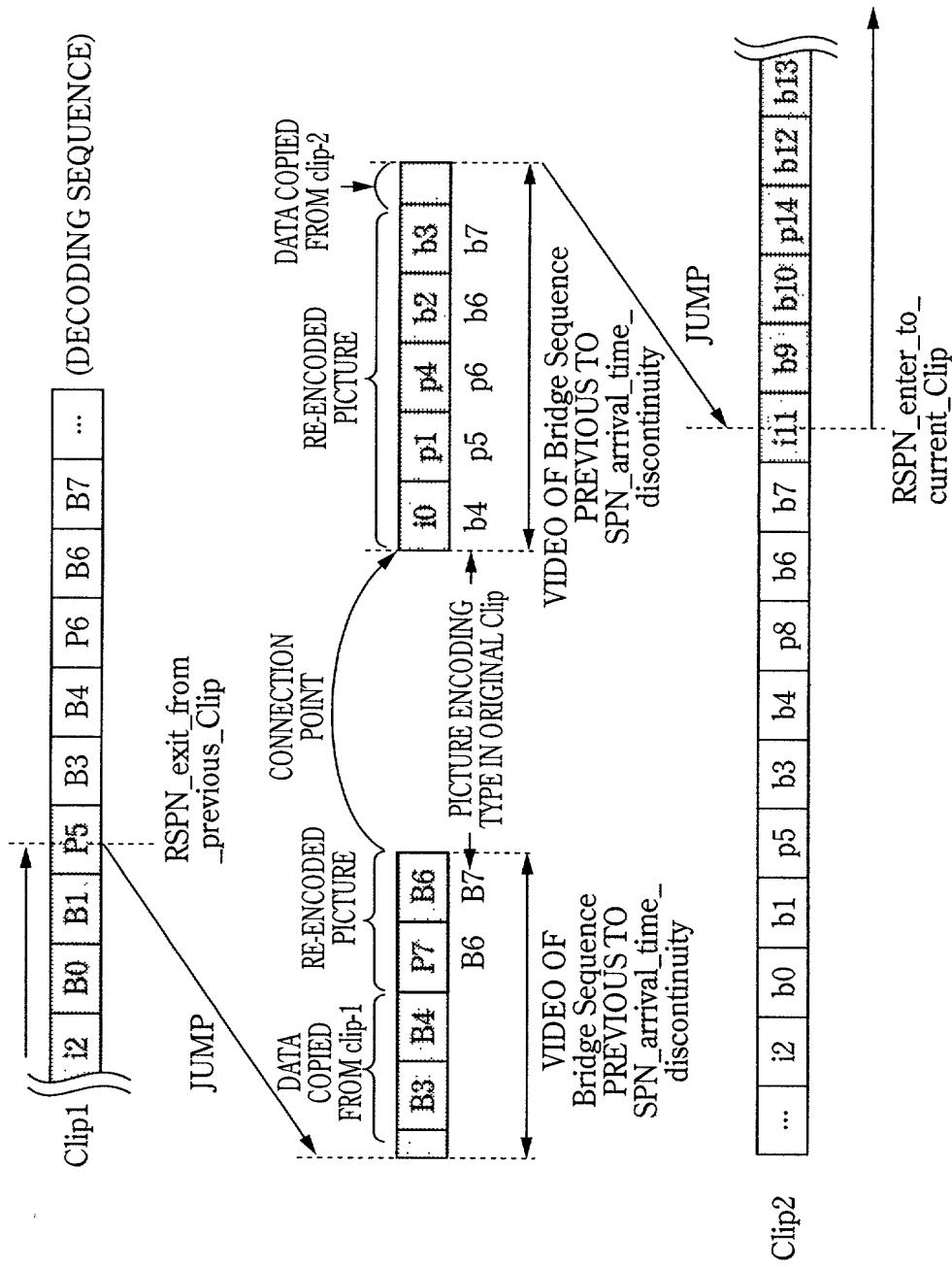


FIG.91

Clip 1: I2 B0 B1 P5 B3 B4 P7 B6 B7

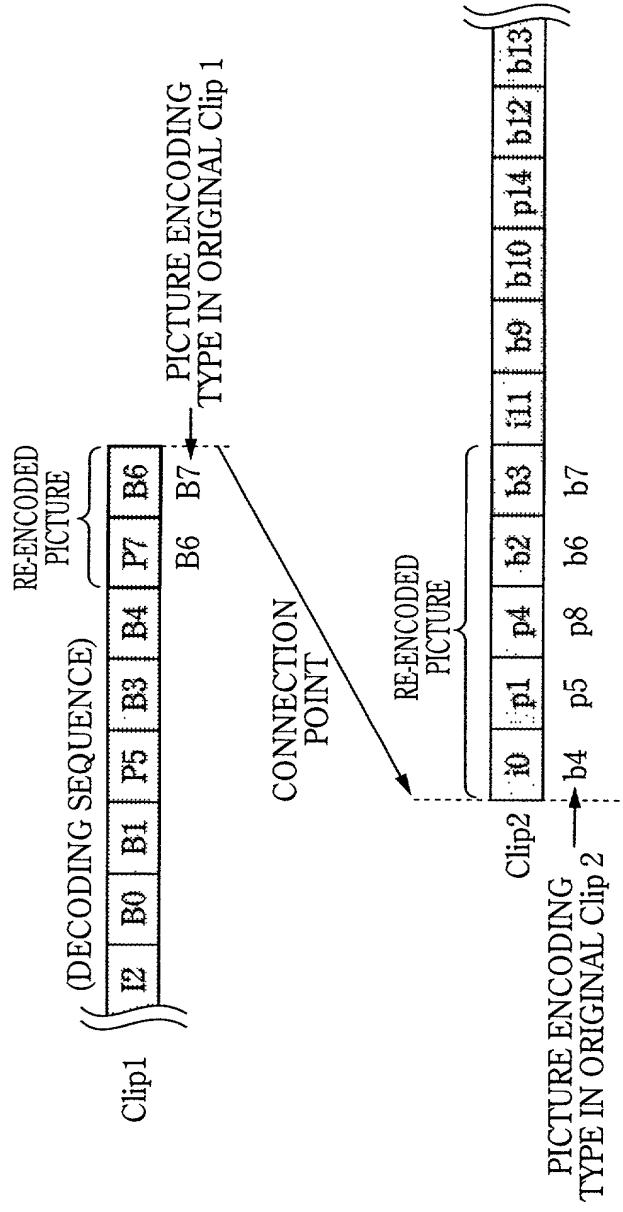


FIG.92

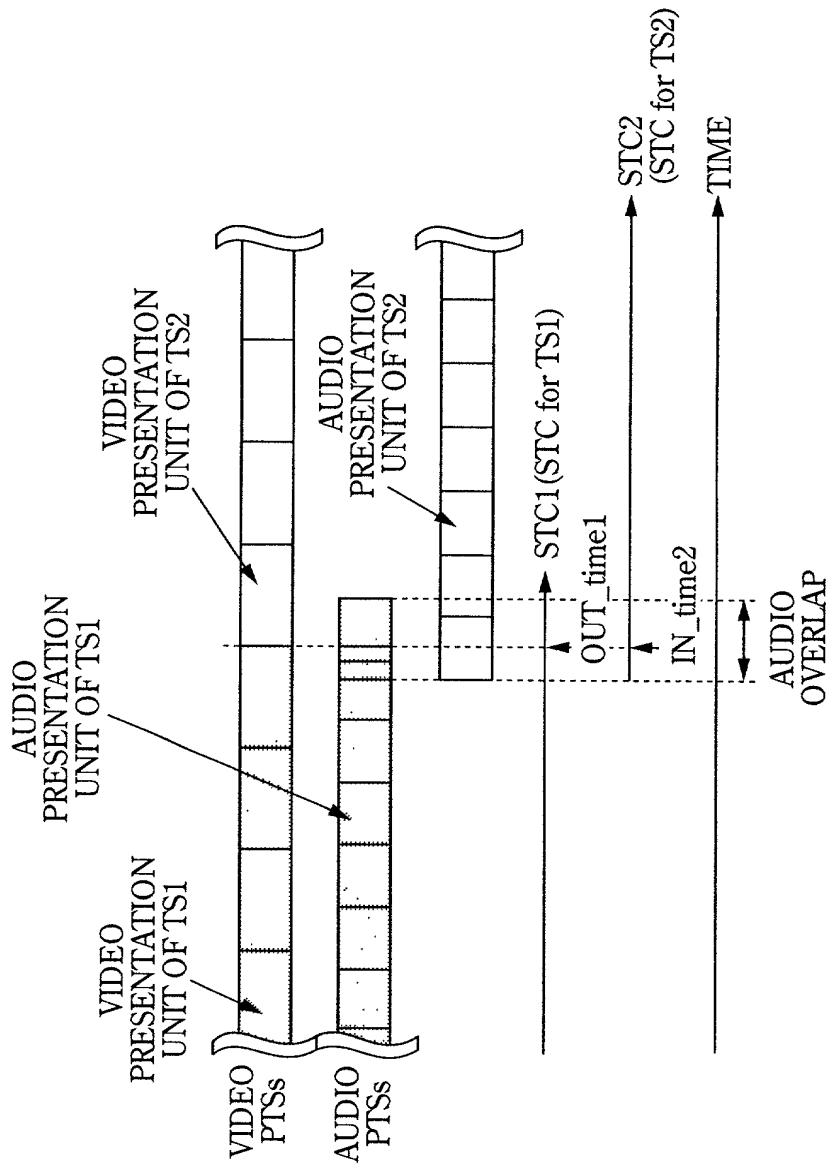


FIG.93

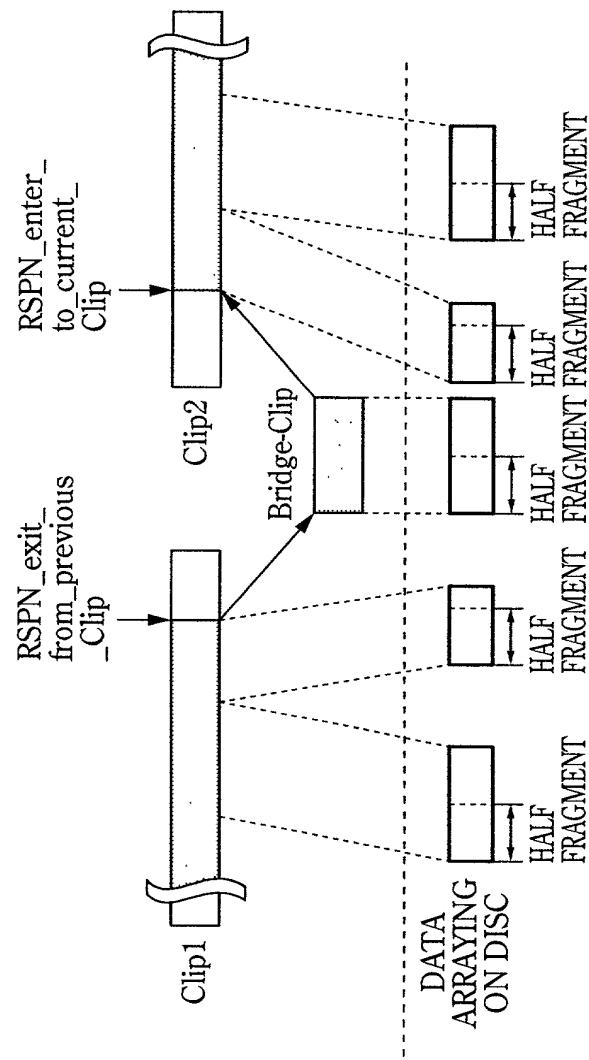


FIG.94

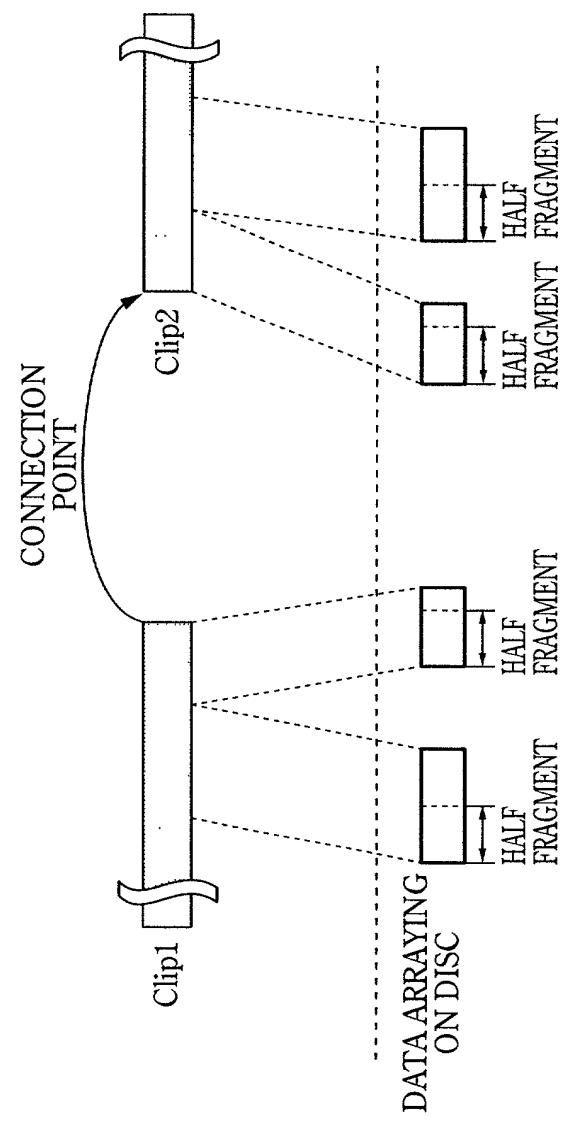


FIG.95

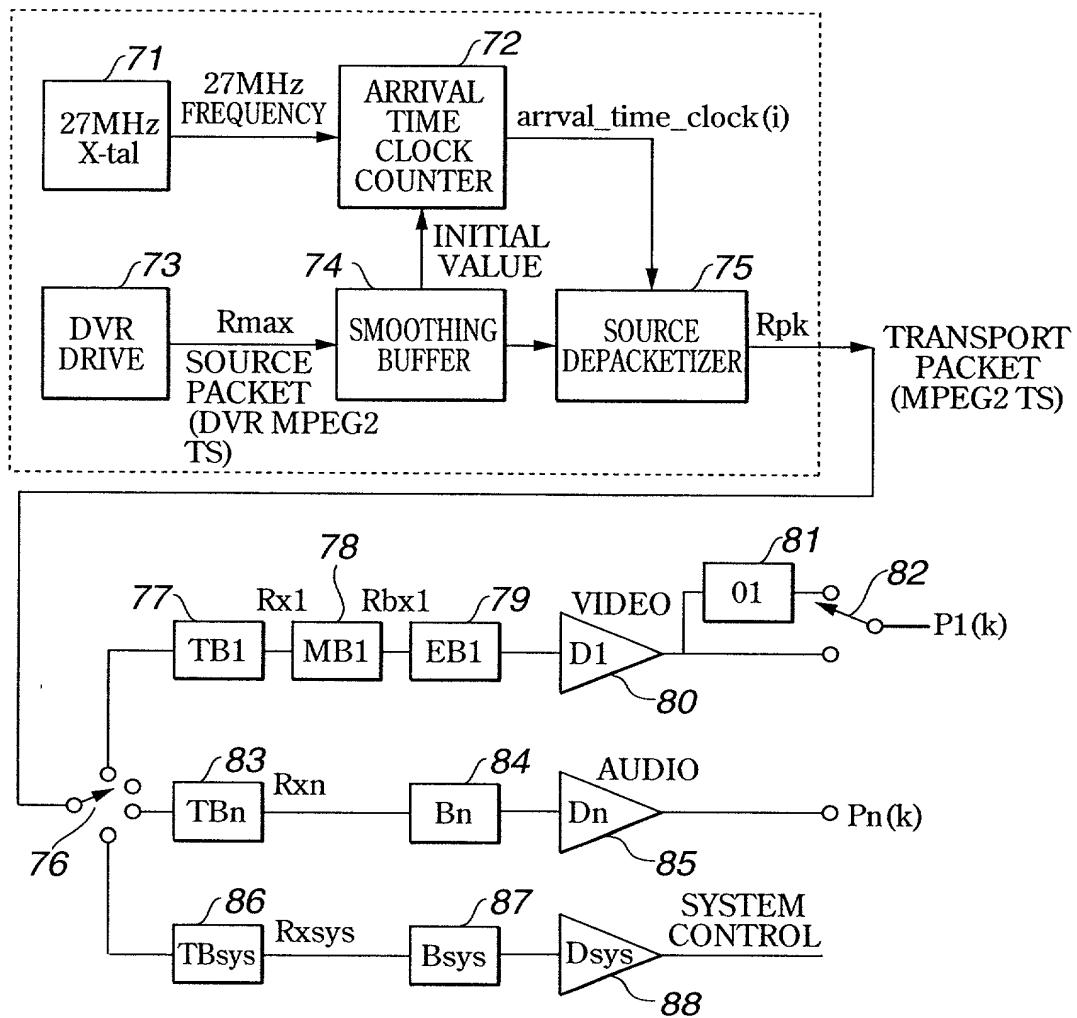


FIG.96

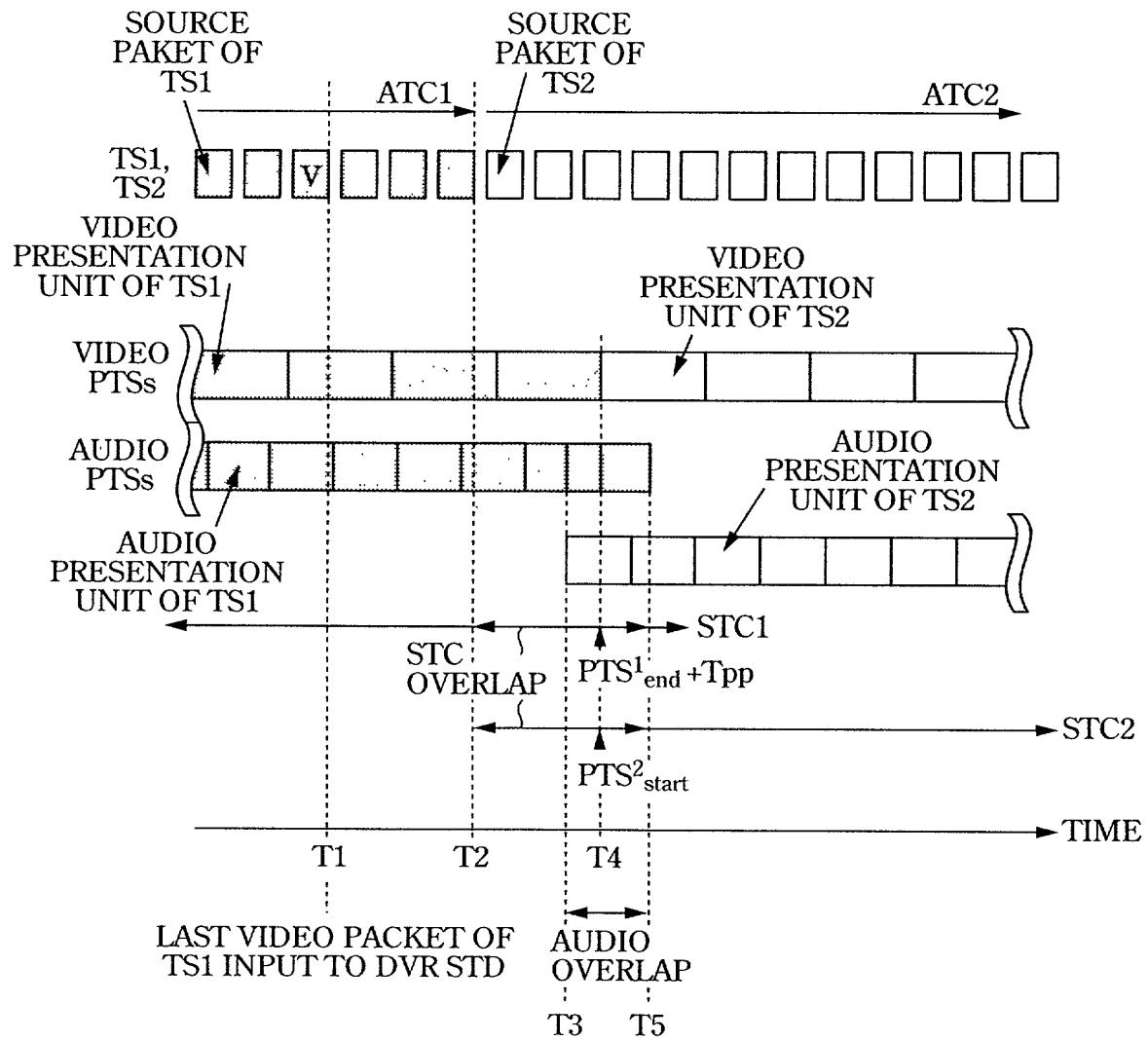


FIG.97

Syntax	No. of bits	Mnemonic
BridgeSequenceInfo()		
Bridge_Clip_Information_file_name	8*10	bslbf
}		

FIG.98

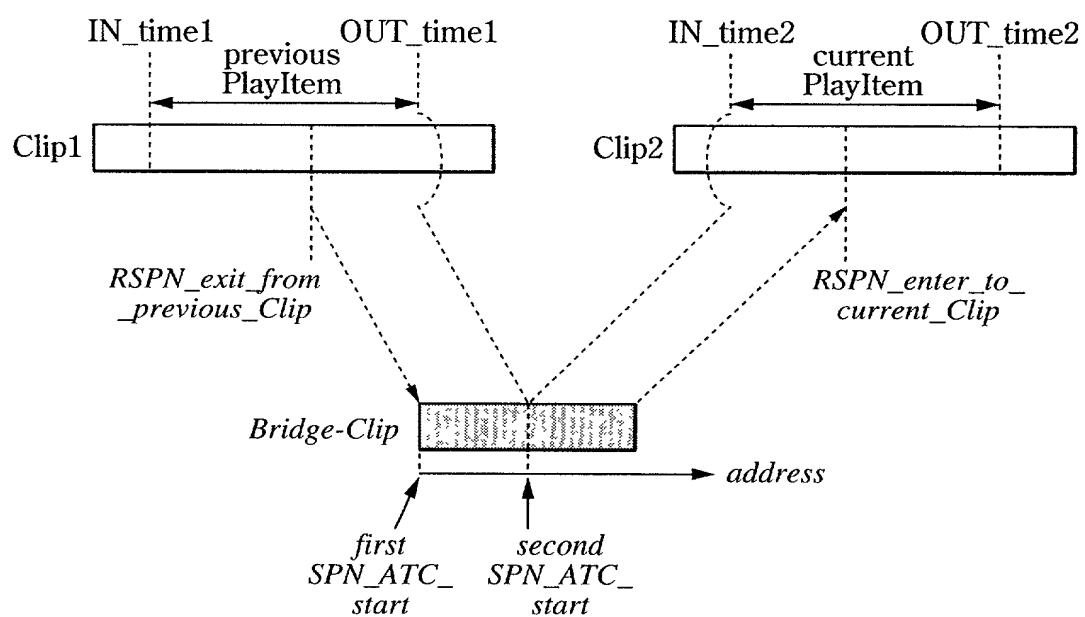


FIG.99

Syntax	No. of bits	Mnemonic
zzzz.clpi {		
version_number	8*4	bslbf
SequenceInfo_start_address	32	uimsbf
ProgramInfo_start_address	32	uimsbf
CPI_start_address	32	uimsbf
ClipMark_start_address	32	uimsbf
MakersPrivateData_start_address	32	uimsbf
reserved_for_future_use	96	bslbf
ClipInfo()		
for (i=0;i<N1;i++) {		
padding_word	16	bslbf
}		
SequenceInfo()		
for (i=0;i<N2;i++) {		
padding_word	16	bslbf
}		
ProgramInfo()		
for (i=0;i<N3;i++) {		
padding_word	16	bslbf
}		
CPI()		
for (i=0;i<N4;i++) {		
padding_word	16	bslbf
}		
ClipMark()		
for (i=0;i<N5;i++) {		
padding_word	16	bslbf
}		
MakersPrivateData()		
for (i=0;i<N6;i++) {		
padding_word	16	bslbf
}		
}		

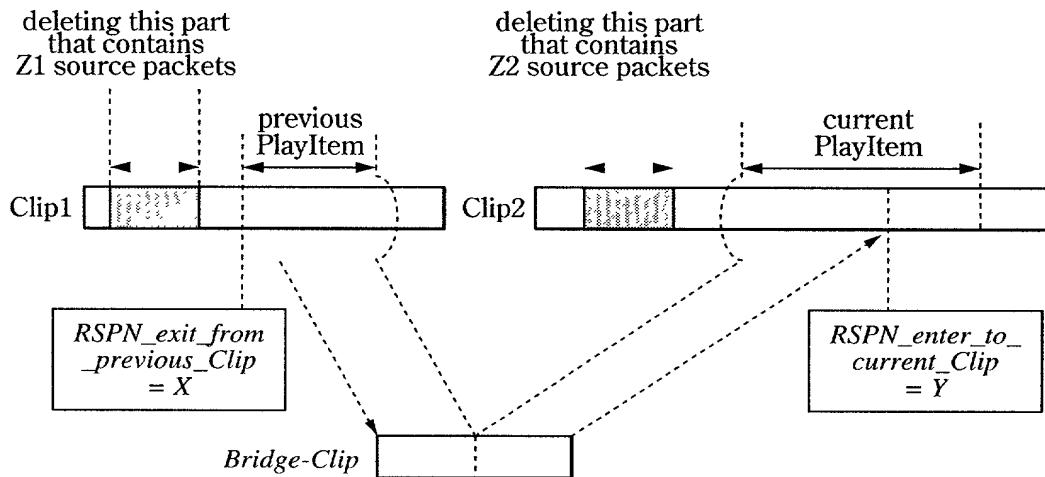
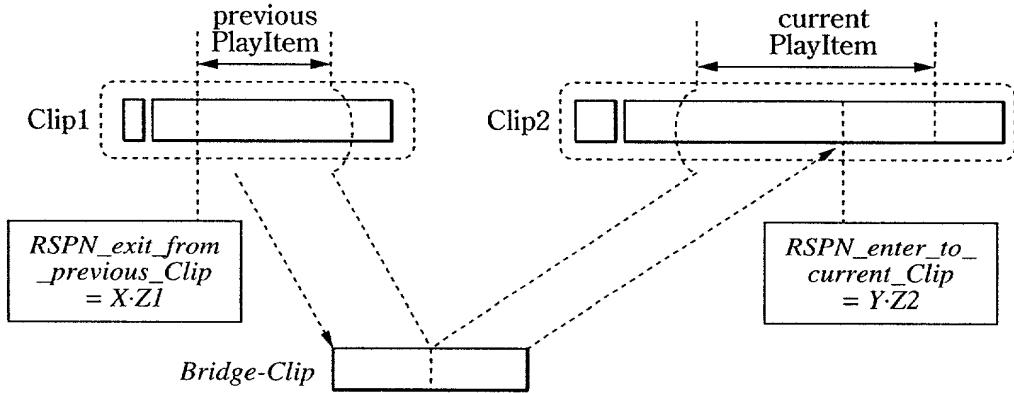
FIG.100

Syntax	No. of bits	Mnemonic
ClipInfo()		
length	32	uimsbf
reserved_for_future_use	16	bslbf
Clip_stream_type	8	uimsbf
reserved_for_word_align	6	bslbf
transcode_mode_flag	1	bslbf
controlled_time_flag	1	bslbf
TS_average_rate	32	uimsbf
TS_recording_rate	32	uimsbf
reserevd_for_DVRsystem_use	144	bslbf
TS_type_info_block()		
if (Clip_stream_type=="Bridge-Clip AV stream") {		
previous_Clip_information_file_name	8*10	bslbf
RSPN_exit_from_previous_Clip	32	uimsbf
current_Clip_information_file_name	8*10	bslbf
RSPN_enter_to_current_Clip	32	uimsbf
)		
}		

FIG.101

Syntax	No. of bits	Mnemonic
SequenceInfo()		
length	32	uimsbf
reserved_for_word_align	8	bslbf
num_of_ATC_sequences	8	uimsbf
for (atc_id=0; atc_id<num_of_ATC_sequences; atc_id++) {		
SPN_ATC_start[atc_id]	32	uimsbf
num_of_STC_sequences[atc_id]	8	uimsbf
offset_STC_id[atc_id]	8	uimsbf
for (stc_id = offset_STC_id[atc_id];		
stc_id		
<(num_of_STC_sequences[atc_id]+offset_STC_id[atc_id];		
stc_id++) {		
PCR_PID[atc_id][stc_id]	16	uimsbf
SPN_STC_start[atc_id][stc_id]	32	uimsbf
presentation_start_time[atc_id][stc_id]	32	uimsbf
presentation_end_time[atc_id][stc_id]	32	uimsbf
}		
}		

FIG.102

Before Editing**FIG.103A**After Editing**FIG.103B**

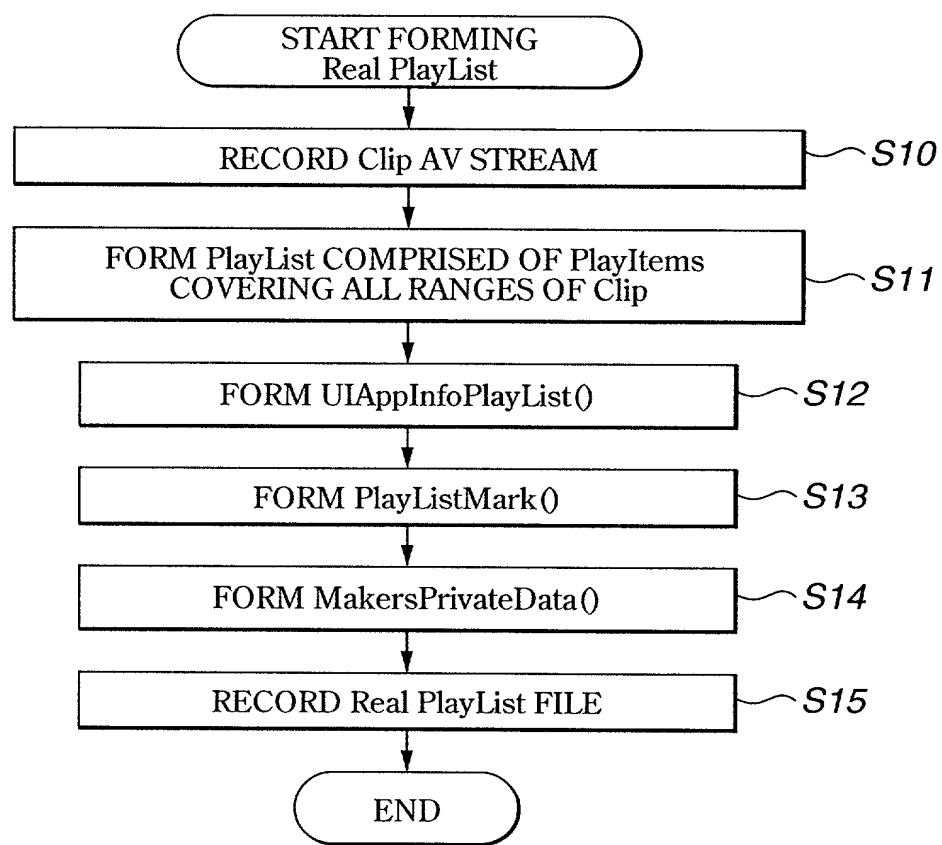


FIG.104

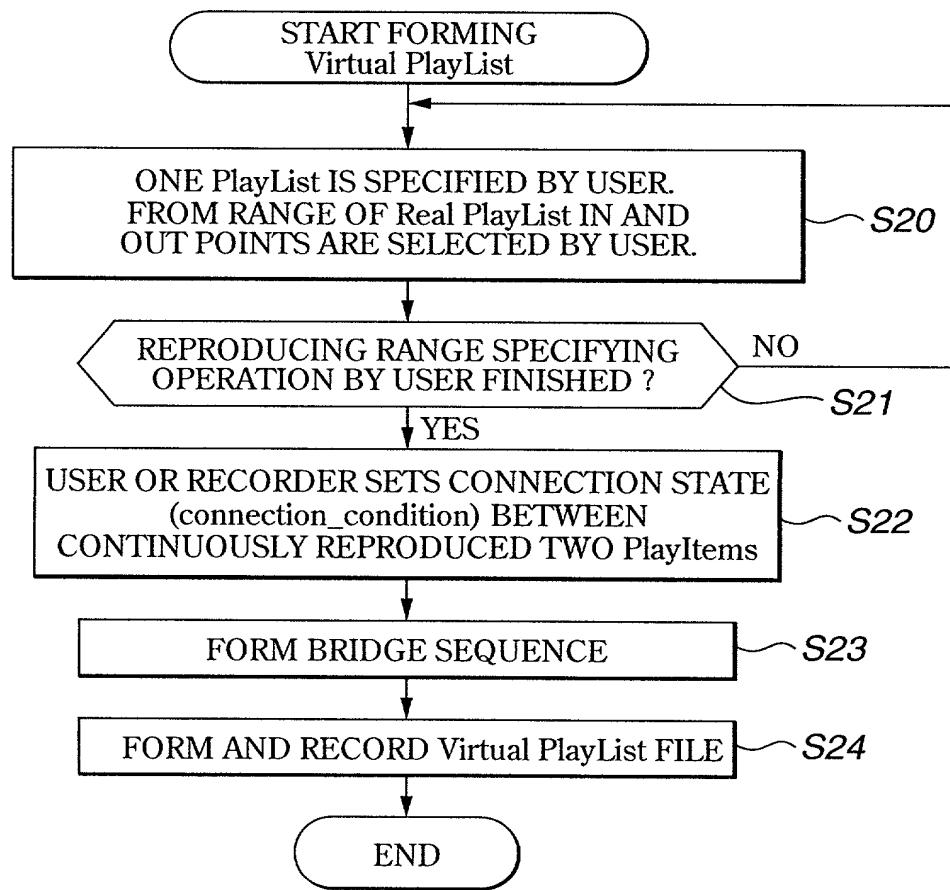


FIG.105

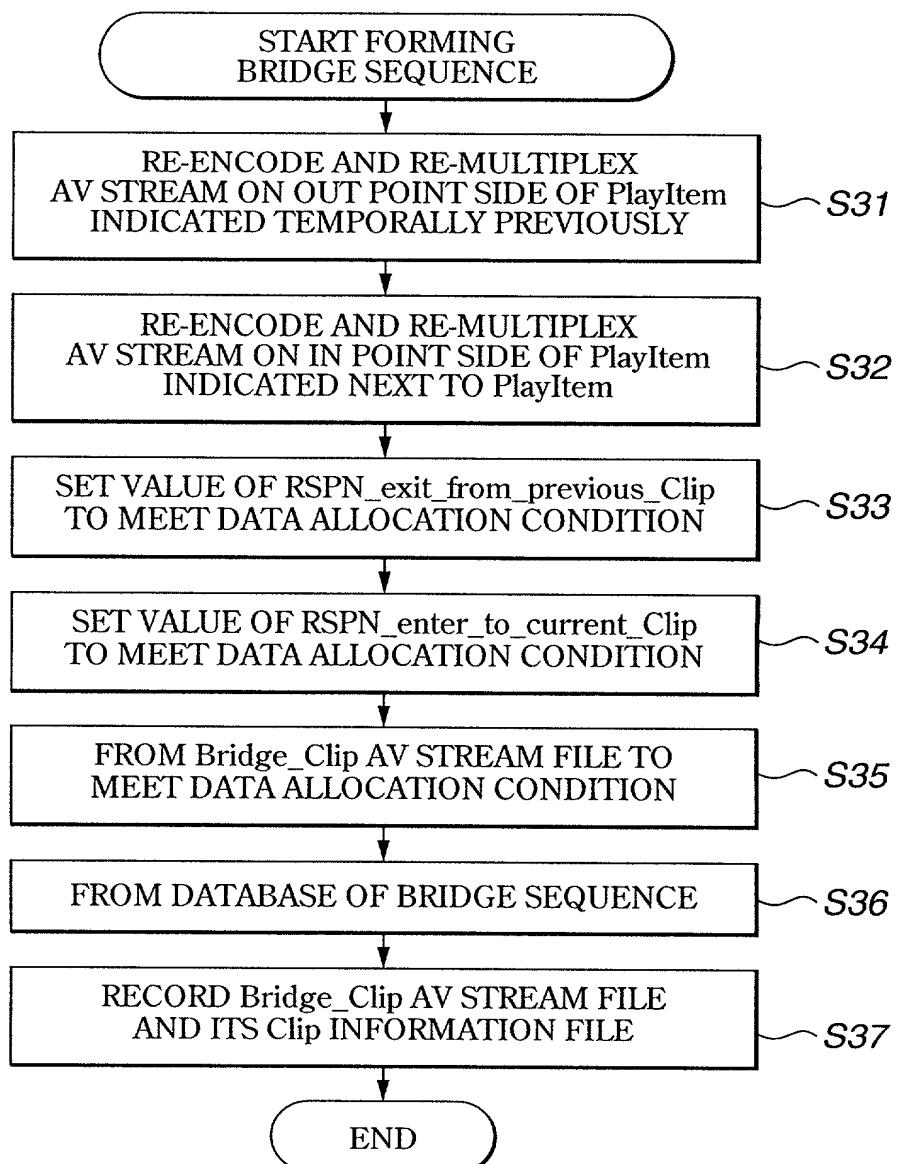


FIG.106

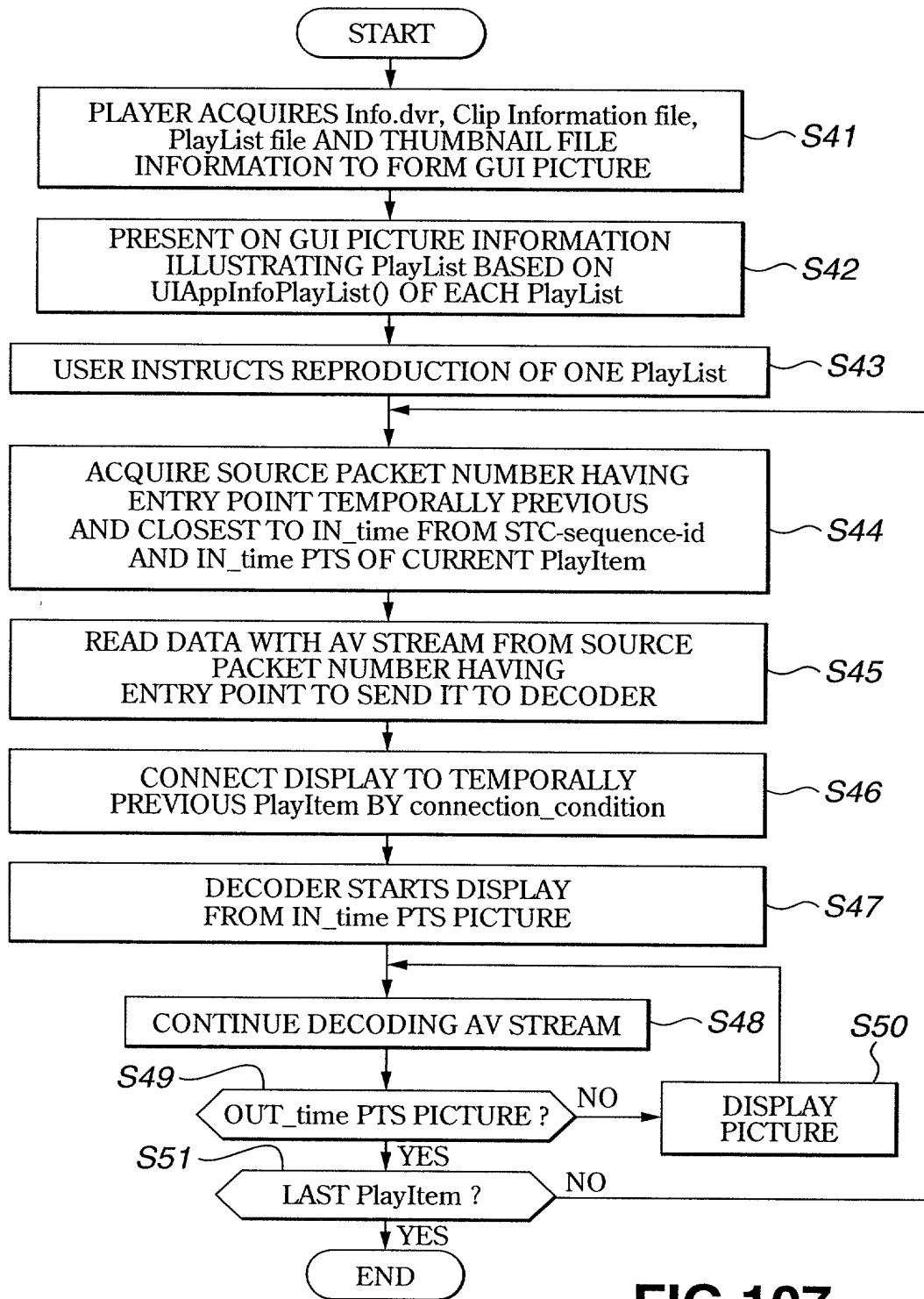


FIG.107

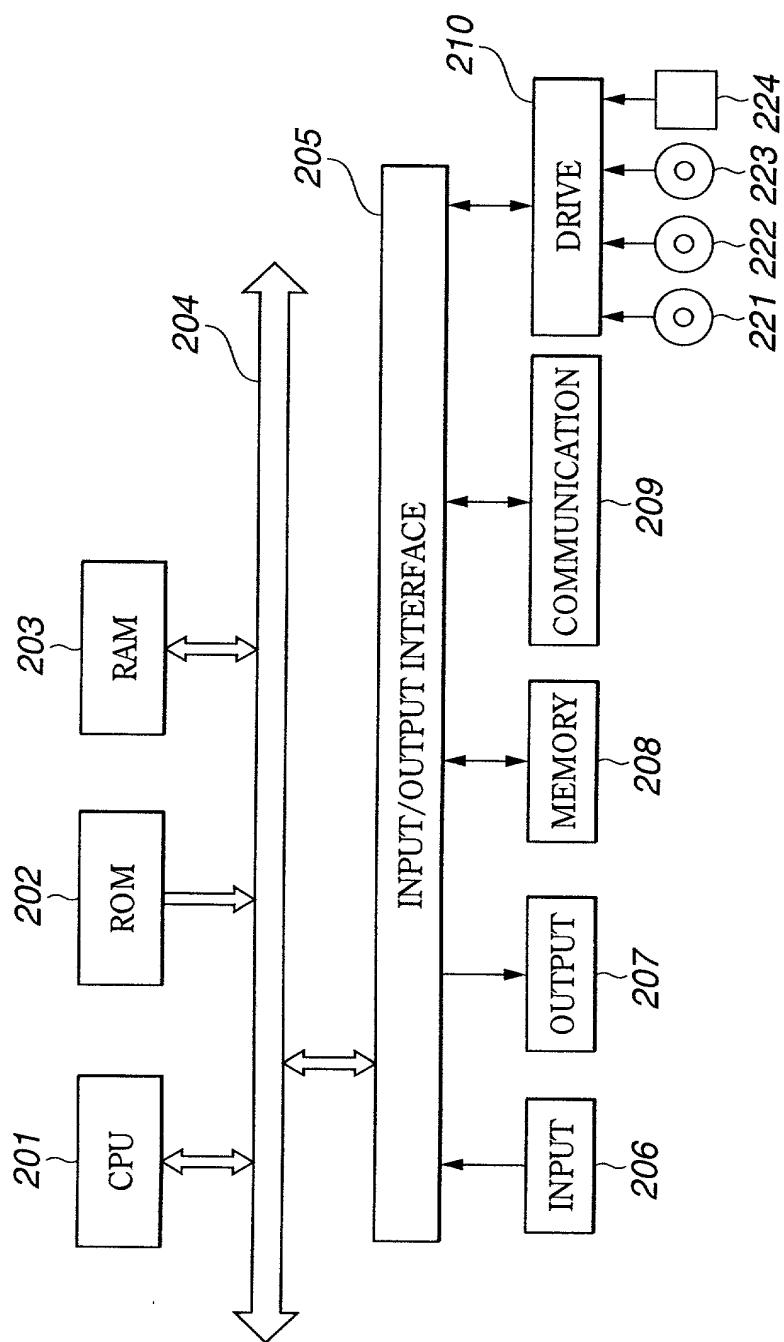


FIG.108